United States Department of Agriculture

Forest Service

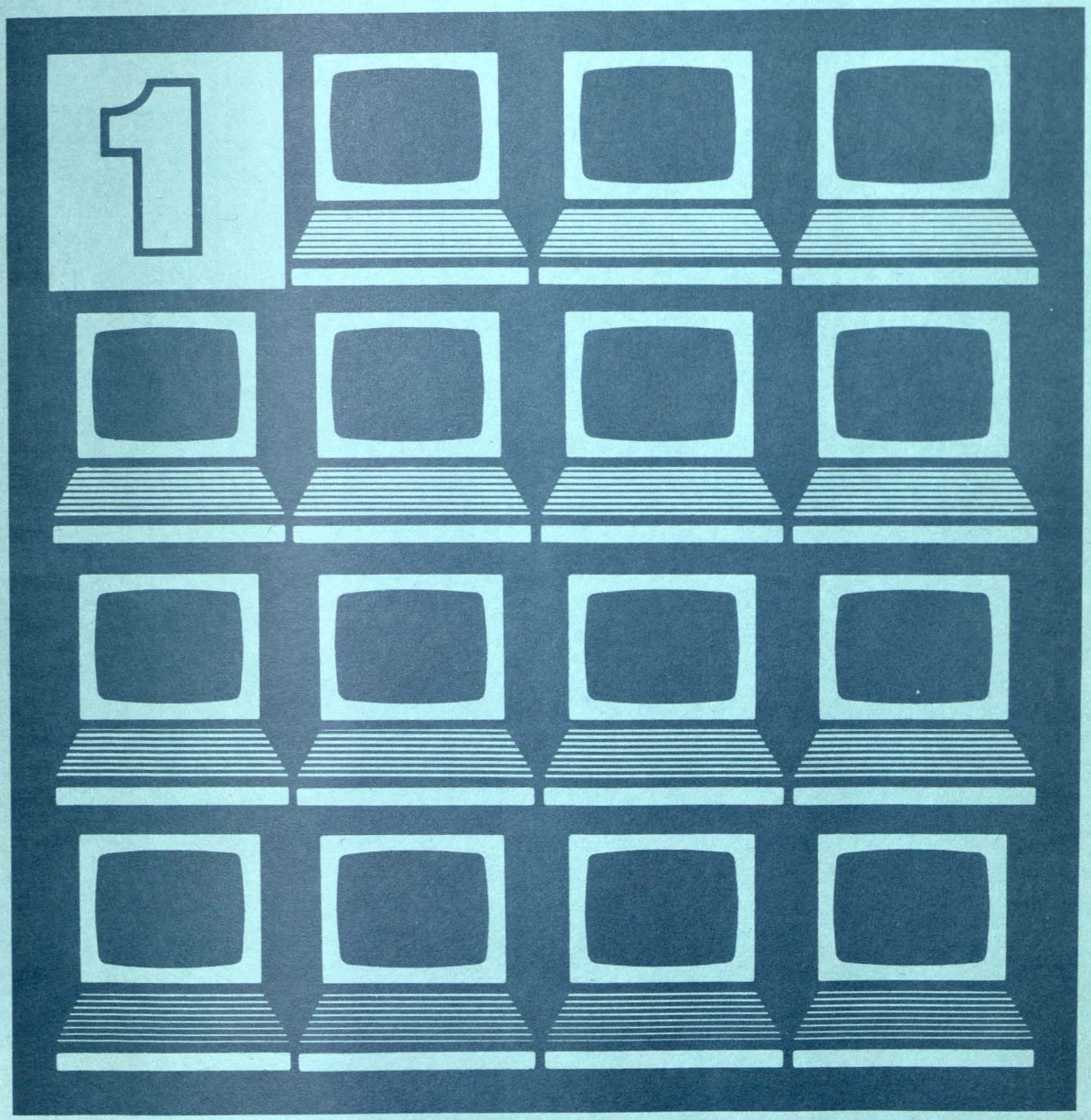
North Central Forest Experiment Station

General Technical Report **NC-84**

The Microcomputer Scientific Software Series 1

The Numerical Information Manipulation System

Harold M. Rauscher



North Central Forest Experiment Station
Forest Service—U.S. Department of Agriculture
1992 Folwell Avenue
St. Paul, Minnesota 55108
Manuscript approved for publication January 5, 1982
1983

PREFACE

The microcomputer scientific software series will present a continuing set of BASIC language, small computer oriented programs, unified by the same data interchange standard. Every program in this series will be input-output compatible with every other program in this series.

Updates, corrections, and changes to programs in this series will be forthcoming as necessary. Users are encouraged to submit extensions, useful changes, or identified errors to the author for inclusion in the next update. Originators will be given full credit for their contributions. It is only through diligent improvement of existing programs that truly outstanding software will eventually become a reality for small computers.

CONTENTS

Page

General Information 1
Identification 1
Summary 1
Quick Reference Guide 2
Application 2
Introduction 2
Operating Instuctions 2
Sample Terminal Session 6
Programming Procedures 15
System Characteristics 15
Data File Structure 16
Program Code Documentation 16
Recovery From Errors 18
Appendix I.—Source code listing for program SEED/BAS 19
Appendix II.—Source code listing for program MAIN/BAS 25
Appendix III.—Source code listing for program ALTER/BAS 30
Appendix IV.—Source code listing for program MATMULT/BAS 35
Appendix V.—Source code listing for program INVERSE/BAS 37
Appendix VI.—Source code listing for program TRNSFORM/BAS 40
Appendix VII.—Source code listing for program ELEMOPS/BAS 42

THE MICROCOMPUTER SCIENTIFIC SOFTWARE SERIES 1: THE NUMERICAL INFORMATION MANIPULATION SYSTEM

Harald M. Rauscher, *Research Forester,*
Marquette, Michigan

GENERAL INFORMATION

Identification

System Name: Numerical Information Manipulation System (NIMS)
Location: North Central Forest Experiment Station
1831 Hwy. 169 E.
Grand Rapids, Minnesota 55744
Language: BASIC
System: Radio Shack TRS-80 Microcomputer[1]

Summary

General

NIMS combines the versatility of a word processing system and the ability to manipulate mathematical or statistical data in numeric matrix form. The system allows operations on single numbers, entire rows or columns of numbers, or on entire data matrices. Data elements may be transformed by logarithmic, power, exponential, and other operations. When data sets are treated as a single matrix, they may be added, subtracted, multiplied, inverted, separated, or merged.

Documentation

This documentation will explain the operation of NIMS, detail how the program functions in enough depth to allow a knowledgeable user to implement improvements and extensions, and present the complete source code listings.

Objectives

(a) Provide small computer users with an efficient and flexible numeric data manipulation system.

(b) Provide users sufficient instructions to help them implement additional capabilities.

Goals

After reading this documentation, users will be able to:

(a) handle NIMS efficiently.
(b) Gain almost total automated control of their numeric data sets.

Assumptions

I assume that the users of NIMS are also programmers. Consequently, this documentation is aimed at those who understand the operating environment of their small computers, can program them in the resident BASIC dialect, and know some of the differences between the many BASIC dialects.

References

Buhyoff, G. J.; Rauscher, H. M.; Hull, IV, R. B.; Killeen, K. Statistical Processing Systems Version 2.0 User's Manual. Blacksburg, VA: Department of Forestry and Forest Products, VPI&SU; 1980. p.

Buhyoff, G .J.; Rauscher, H. M.; Hull, IV, R. B.; Killeen, K. Microcomputer-resident comprehensive statistical analysis. Behavior Methods and Instrumentation 12: 551-553; 1980.

Hornbeck, R. W. Numerical Methods. New York, NY: Quantum Publishing, Inc.; 1975, 310 p.

[1]*Mention of trade names does not constitute endorsement of the products by the USDA Forest Service.*

Rauscher, H. M.; Buhyoff, G. J. Utility subroutines for data manipulations. Behavior Research Methods and Instrumentation; Instrumentation 14: 37-38; 1981.

Rowlett, F. B. Menu List Selection Subroutine. 80-Microputing 1: 195-196; 1980.

Quick Reference Guide

The functions performed by the data management system are:

1. Matrix alteration program
   (a) change any one element
   (b) delete a column
   (c) delete a row
   (d) add a column
   (e) add a row
   (f) add a row/column with a fixed value
   (g) set any portion of a row/column to a fixed value
   (h) merge any two matrices either by rows or by columns
   (i) extract any portion of a matrix as a new matrix
2. Add or subtract any two matrices
3. Multiply all elements of a matrix by a scalar
4. Multiply two matrices together
5. Transpose a matrix
6. Invert a matrix
7. Transform the elements of a matrix to
   (a) absolute values
   (b) integer values
   (c) exponential values
   (d) elements raised to a power
   (e) square roots
   (f) natural logarithms
   (g) base 10 logarithms
   (h) smooth variables by moving averages
8. Create a diagonal matrix
9. Perform row/column operations
   (a) interchange two rows
   (b) interchange two columns
   (c) multiply a row by a constant
   (d) multiply a column by a constant
   (e) add one row multiplied by a constant to another row
   (f) add one column multiplied by a constant to another column
   (g) divide a row/column by any element in the matrix by referring to its row/column identifier

APPLICATION

Introduction

This section of the guide explains how to use NIMS. It presents each option in the system and then illustrates it in a sample terminal session.

Operating Instructions

Once the computer system is under control of the BASIC interpreter, the main program of NIMS is executed by typing <RUN> "MAIN/BAS". A menu of choices, numbered 1-16, appears on the video display and you are asked to make a selection (fig. 1). The first seven selections remain the same for all menus. Selection <1> returns you to the menu for the main program regardless of where you are in the system. Selection <2> causes the NIMS system to relinquish control to the BASIC interpreter. Choices 3-7 read a matrix from disk, write a matrix to disk, allow keyboard entry to data, display a data matrix, and print a data matrix to a lineprinter, respectively. See documentation by Rauscher and Buhyoff (1981) for further details.

The video display of matrix A(R,C) is a window that shows a block of data (13 rows by 4 columns) on the screen. By depressing the up, down, left, or right arrows on the keyboard, you select a portion of the data matrix for display. Pressing the <.> key activates a subroutine to change the names of the variables of the data set. Pressing the <CLEAR> key returns you to the menu section of the current program. After completion of the subroutines activated by choices 3-7, the main menu is displayed again.

Choice <8> executes the matrix alteration program. The matrix alteration menu is displayed (fig. 2) and you are asked to select from options 1-15. Selections 1-7, as discussed above, remain the same. Selection <8> causes the program to ask for the row and column identification of the element to be changed. When a coordinate pair is entered, say <4,5>, the program displays the current value of the matrix element (row = 4, column = 5) and prompts for the new value. When you provide the new value, the program displays the matrix on the video screen to confirm that the change had indeed been made. To preserve the change, you must select <4>, which activates the routine to save the data on disk. WARNING! If you fail to save the current data set to disk when the program cautions you to do so, you may lose the data set.

FORESTRY SCIENCES LABORATORY, U.S. FOREST SERVICE
MENU FOR: NUMERICAL INFORMATION MANIPULATION SYSTEM

1. RETURN TO MAIN MENU
2. GO TO BASIC
3. READ A MATRIX FROM DISK
4. WRITE A MATRIX TO DISK
5. ENTER A MATRIX MANUALLY
6. DISPLAY MATRIX ON VIDEO
7. PRINT MATRIX
8. ALTER MATRIX
9. ADD/SUB TWO MATRICES
10. SCALAR/MATRIX OPERATIONS
11. MATRIX MULTIPLY
12. TRANSPOSE A MATRIX
13. INVERSE A MATRIX
14. TRANSFORM DATA
15. CREATE SPECIAL MATRICES
16. ROW/COL OPERATIONS

SELECT ONE OF THE ABOVE—>.

Figure 1.—*Menu of choices for program MAIN of NIMS.*

Option <9> deletes any column and <10> deletes any row in the data matrix. Option <11> adds one column to the end of the data matrix. A column may not be added between two other columns. Option <12> adds one row to the end of the matrix. A row may not be added between two other rows. Under options <11> and <12>, you must put in one element of the row/column at a time. Option <13> adds a row/column where all elements are the same value. Option <14> allows portions of any row or column to be set to a specific fixed value.

Selection <15> merges two matrices. Before entering this option, you must put the base matrix into memory by choosing option <3> or <5>. The (1,1) element of the base matrix will remain as the (1,1) element of the new matrix. You are prompted to enter the name of the second matrix, i.e., the one that will be moved next to the base matrix. The program then prompts you to select merging by rows or columns. The rows or columns of the second matrix are added to the rows or columns of the base matrix to form the new matrix. Zero elements are added where the dimensions of the two matrices do not match. The merging procedure will become clear with a little experimentation.

Selection <16> extracts a submatrix from the original matrix. Any block of contiguous data elements may be removed from any matrix by specifying the coordinates of the upper left hand corner and those of the lower right hand corner. Selection <1> returns you to the main menu.

Selection <9> from the main menu (fig. 1) executes the add/subtract routine for two matrices. Matrix addition and subtraction is defined only for matrices with the same dimensions. You are prompted

FORESTRY SCIENCES LABORATORY, U.S. FOREST SERVICE
MENU FOR: MATRIX ALTERATION ROUTINE

1. RETURN TO MAIN MENU
2. GO TO BASIC
3. READ A MATRIX FROM DISK
4. WRITE A MATRIX TO DISK
5. ENTER A MATRIX MANUALLY
6. DISPLAY MATRIX ON VIDEO
7. PRINT MATRIX
8. CHANGE AN ELEMENT
9. DELETE A COLUMN
10. DELETE A ROW
11. ADD A COLUMN
12. ADD A ROW
13. ADD FIXED VALUE ROW/COL
14. CHANGE R/C TO FIXED VAL
15. MERGE TWO MATRICES
16. SPLIT A MATRIX

SELECT ONE OF THE ABOVE—>.

Figure 2.—*Menu of choices for program ALTER of NIMS.*

to enter matrix B into the memory and then to enter matrix C. The next prompt requests a choice between operations B+C and B-C. When the pertinent calculations are completed, the main menu is once again displayed.

Option <10> allows all elements in a matrix to be multiplied by a scalar (fixed number) or it allows a scalar to be added to all elements. Elements can be divided by using the inverse of the scalar and subtracted by using its negative.

Option <11> activates the matrix multiplication routine (fig. 3) and option <8> from the matrix multiplication menu allows multiplication of two matrices. The order of matrix multiplication is B x A where B is entered first and A second. The two matrices must be conformable under multiplication. Option <1> returns control to the main menu.

Selection <12> (fig. 1) executes the matrix transposition subroutine. You are prompted to enter the matrix, the rows and columns are transposed, and control is returned to the main menu.

Option <13> leads to the matrix inversion menu (fig. 4). Option <8> activates the matrix inversion routine. This routine uses a Gauss-Jordan column elimination algorithm after Hornbeck (1975). Only nonsingular, square matrices may be inverted with this algorithm. If the matrix is singular, an error trap tells you of this fact and shifts program control to the matrix inversion menu. Otherwise, the matrix is inverted, the determinant is displayed, and control is returned to the matrix inversion menu. The matrix inversion routine in this program uses single precision variables (6-digit accuracy), but this may not be enough to accurately invert those matrices that approach singularity. No attempt has been made to deal with this problem of error analysis. Caution: you should check the accuracy of the inversion by multiplying the original matrix by its inverse to see how well the identity matrix is achieved.

Returning to the main menu (fig. 1), option <14> leads to the data transformation matrix (fig. 5). Options 8-14 are straightforward transformations of every element in the data matrix. Option <1> returns control to the main menu.

Option <15> (fig. 1) creates a diagonal matrix. This saves time because only the diagonal elements of the matrix need to be explicitly entered. The off-diagonal zeroes are automatically assigned. Under the same option, a matrix of fixed values may also be quickly created. For a sparse data set, you may want to create a matrix of zeroes and then change the elements that are nonzero.

FORESTRY SCIENCES LABORATORY, U.S. FOREST SERVICE
MENU FOR: MATRIX MULTIPLICATION ROUTINE

1. RETURN TO MAIN MENU
2. GO TO BASIC
3. READ A MATRIX FROM DISK
4. WRITE A MATRIX TO DISK
5. ENTER A MATRIX MANUALLY
6. DISPLAY MATRIX ON VIDEO
7. PRINT MATRIX
8. MATRIX MULTIPLY

SELECT ONE OF THE ABOVE—>.

Figure 3.—*Menu of choices for program MATMULT of NIMS.*

FORESTRY SCIENCES LABORATORY, U.S. FOREST SERVICE
MENU FOR: MATRIX INVERSION ROUTINE

1. RETURN TO MAIN MENU
2. GO TO BASIC
3. READ A MATRIX FROM DISK
4. WRITE A MATRIX TO DISK
5. ENTER A MATRIX MANUALLY
6. DISPLAY MATRIX ON VIDEO
7. PRINT MATRIX
8. MATRIX INVERSION

SELECT ONE OF THE ABOVE—>.

Figure 4.—*Menu of choices for program INVERSE of NIMS.*

FORESTRY SCIENCES LABORATORY, U.S. FOREST SERVICE
MENU FOR: MATRIX DATA TRANSFORMATION ROUTINE

1. RETURN TO MAIN MENU
2. GO TO BASIC
3. READ A MATRIX FROM DISK
4. WRITE A MATRIX TO DISK
5. ENTER A MATRIX MANUALLY
6. DISPLAY MATRIX ON VIDEO
7. PRINT MATRIX
8. ABSOLUTE VALUE
9. INTEGER VALUE
10. EXPONENTIATION
11. POWERS
12. SQUARE ROOT
13. NATURAL LOGARITHM
14. BASE 10 LOGS

SELECT ONE OF THE ABOVE—>.

Figure 5.—*Menu of choices for program TRNSFORM of NIMS.*

FORESTRY SCIENCES LABORATORY, U.S. FOREST SERVICE
MENU FOR: ELEMENTARY ROW/COLUMN OPERATIONS ROUTINE

1. RETURN TO MAIN MENU
2. GO TO BASIC
3. READ A MATRIX FROM DISK
4. WRITE A MATRIX TO DISK
5. ENTER A MATRIX MANUALLY
6. DISPLAY MATRIX ON VIDEO
7. PRINT MATRIX
8. FLIP TWO ROWS
9. FLIP TWO COLUMNS
10. ROW x SCALAR
11. COLUMN x SCALAR
12. ROW + SCALAR x ROW
13. COL + SCALAR x COL
14. (ROW OR COL) / ELEMENT

SELECT ONE OF THE ABOVE—>.

Figure 6.—*Menu of choices for program ELEMOPS of NIMS.*

Option <16> activates the elementary row/column operations menu (fig. 6). Elementary operations are carefully defined by the theory of linear algebra. A matrix that has been changed only through elementary operations is equivalent to its original form. The three allowable elementary operations are:

a. Interchange of rows/columns;
b. Multiplication of rows/columns by a scalar not equal to zero;
c. Addition of a row/column multiplied by a scalar to another row/column.

Options 8-13 (fig. 6) are self-explanatory. Option <14> simplifies the use of elementary operation b. You can make the leading element of a row equal to 1 by multiplying the entire row by the inverse of the first element in that row. This operation, equivalent to dividing the row by its leading element, is allowed in option <14> by specifying the row/column coordinates of the element.

Sample Terminal Session

This sample terminal session illustrates most of the operations discussed earlier.

```
                              (Beginning)

          NORTHERN HARDWOODS LABORATORY, U.S. FOREST SERVICE

        MENU FOR:  DATA MANAGEMENT SYSTEM

 1.  RETURN TO MAIN MENU                  2.  GOTO BASIC
 3.  READ A MATRIX FROM DISK              4.  WRITE A MATRIX TO DISK
 5.  ENTER A MATRIX MANUALLY              6.  DISPLAY A MATRIX ON VIDEO
 7.  PRINT MATRIX                         8.  ALTER MATRIX
 9.  ADD/SUB TWO MATRICES                10.  SCALAR/MATRIX OPERATIONS
11.  MATRIX MULTIPLY                     12.  TRANSPOSE A MATRIX
13.  INVERSE OF A MATRIX                 14.  TRANSFORM DATA
15.  CREATE SPECIAL MATRICES             16.  ROW/COL OPERATIONS

 SELECT ONE OF THE ABOVE--> 5.

PLEASE ENTER:  NO. ROWS AND COLS (R,C).--> ? 2,3

ENTER THE NAME FOR VARIABLE 1 --> ONE.....
ENTER THE NAME FOR VARIABLE 2 --> TWO.....
ENTER THE NAME FOR VARIABLE 3 --> THREE...

ROW 1
PLEASE ENTER:  VALUE OF COLUMN 1 --> 13.......
PLEASE ENTER:  VALUE OF COLUMN 2 --> 67.......
PLEASE ENTER:  VALUE OF COLUMN 3 --> 1,34.....
ROW 2
PLEASE ENTER:  VALUE OF COLUMN 1 --> 1005.....
PLEASE ENTER:  VALUE OF COLUMN 2 --> 43.......
PLEASE ENTER:  VALUE OF COLUMN 3 --> 92.......

          NORTHERN HARDWOODS LABORATORY, U.S. FOREST SERVICE

        MENU FOR:  DATA SYSTEM MANAGEMENT

 1.  RETURN TO MAIN MENU                  2.  GOTO BASIC
 3.  READ A MATRIX FROM DISK              4.  WRITE A MATRIX TO DISK
 5.  ENTER A MATRIX MANUALLY              6.  DISPLAY A MATRIX ON VIDEO
 7.  PRINT MATRIX                         8.  ALTER MATRIX
 9.  ADD/SUB TWO MATRICES                10.  SCALAR/MATRIX OPERATIONS
11.  MATRIX MULTIPLY                     12.  TRANSPOSE A MATRIX
13.  INVERSE OF A MATRIX                 14.  TRANSFORM DATA
15.  CREATE SPECIAL MATRICES             16.  ROW/COL OPERATIONS

 SELECT ONE OF THE ABOVE --> 6.
```

Execution begins

First Display on Video Screen

Type input after an arrow or question mark prompt by the program.

Originate a matrix by selecting #5 and entering the example data set into its matrix structure.

After completing each task, you are returned to the command MENU for further instructions.

To look at a data matrix, select #6.

```
COL:     ONE          TWO          THREE
-------------------------------------
R 1:     13           67           1.34
R 2:     1005         43           92
```

The example data set just entered

While inside option #6, depressing the period "." will allow changing variable names. Rows cannot be named.

```
WHICH COLUMN'S NAME WILL BE CHANGED? --> 2.
PLEASE ENTER NEW NAME --> VAR. 02.
```

```
COL:     ONE          VAR. 02      THREE
-------------------------------------
R 1:     13           67           1.34
R 2:     1005         43           92
```

Data set names must be less than or equal to 7 letters.

Each data set must be saved to disk, selection #4, to preserve it. Caution: Failure to do this can mean loss of a data set.

```
ENTER FILENAME OF NEW DATA SET --> ? TEST

ENTER DISK DRIVE # WHERE DATA SHOULD BE SAVED --> ? 1_
```

Once the example set, now called "TEST", is saved to disk, you may retrieve it into active memory at any time through selection #3.

```
ENTER FILENAME OF DESIRED DATA SET --> ? TEST_
```

```
PRESS <ENTER> WHEN PRINTER IS READY.?

PLEASE ENTER NAME OF DATA SET --> TEST.....
```

If a lineprinter is available, you may print TEST to obtain a paper copy.

```
                TEST

COL:     ONE          VAR. 02      THREE
-------------------------------------
R 1:     13           67           1.34
R 2:     1005         43           92
```

Lineprinter output.

Selection #10 allows scalar/matrix operations. You may multiply by a scalar or add a scalar value to a data set.

```
ENTER (M)ULTIPLY OR (A)DD/SUBTRACT BY A SCALAR VALUE --> ? M_
```

You may subtract by adding a negative number.

```
ENTER VALUE OF SCALAR MULTIPLIER --> ? 100_

OPERATE ON A SINGLE (R)OW, (C)OLUMN OR (A)LL ELEMENTS --> ? C_
```

```
ENTER COLUMN NUMBER TO OPERATE ON --> ? 2_

COL:     ONE        VAR. 02      THREE
-----------------------------------------
R 1:     13           6700        1.34
R 2:     1005         4300        92
```

Results of multiplying column #2 by 100.

```
COL:      1          2
-------------------------------
R 1:      13         1005
R 2:      6700       4300
R 3:      1.34       92
```

To transpose a matrix, select #12. This procedure will destroy the variable names previously assigned to the columns. Remember, to preserve this new matrix, you must write it to disk via selection #4.

```
THIS ROUTINE ADDS OR SUBTRACTS MATRICES.

PLEASE ENTER:  (A)DD MATRIX B+C OR (S)UBTRACT B-C --> A

ENTER MATRIX B FROM (1) DISK (2) KEYBOARD OR (3) ALREADY

IN MEMORY --> 3

PLEASE ENTER: C FROM 1=DISK, 2=KEYBOARD, 3=ALREADY LOADED? 3_
```

To add or subtract active matrices, choose selection #9.

```
COL:     ONE        VAR. 02      THREE
-----------------------------------------
R 1:     26           134         2.68
R 2:     2010         86          184
```

Adding TEST to itself will double the values in the matrix.

Selection #8 erases the main program and executes a new one. On the new menu list, selections 1-7 have not changed.

```
          NORTHERN HARDWOODS LABORATORY, U.S. FOREST SERVICE

                MENU FOR:  MATRIX ALTERATION ROUTINE

 1.  RETURN TO MAIN MENU                2.  GOTO BASIC
 3.  READ A MATRIX FROM DISK            4.  WRITE A MATRIX TO DISK
 5.  ENTER A MATRIX MANUALLY            6.  DISPLAY MATRIX ON VIDEO
 7.  PRINT MATRIX                       8.  CHANGE AN ELEMENT
 9.  DELETE A COLUMN                   10.  DELETE A ROW
11.  ADD A COLUMN                      12.  ADD A ROW
13.  ADD FIXED VALUE ROW/COL           14.  CHANGE R/C TO FIXED VAL
15.  MERGE TWO MATRICES                16.  SPLIT A MATRIX

 SELECT ONE OF THE ABOVE--> 8.
```

```
PLEASE ENTER:  LOCATION OF ELEMENT TO BE CHANGED (ROW, COL).? 1,3

MATRIX (R,C)= ( 1 , 3 )= 1.34

PLEASE ENTER:  NEW VALUE OF ELEMENT.? -99.99_
```

To change any one element, you must first enter the desired data set via selection #3 or #5 then selection #8 will change any single element.

```
COL:     ONE         VAR. 02       THREE
-----------------------------------------
R 1:     13            67         -99.99
R 2:     1005          43           92
```

Notice the change in data set TEST. The number -99.99 is used to identify missing values.

```
PLEASE ENTER:  COLUMN TO BE DELETED.? 2_
```

Selection #9 deletes any column and #10 deletes any row.

```
COL:     ONE         THREE
-----------------------------
R 1:     13          -99.99
R 2:     1005         92
```

Column #2 removed from data set TEST.

```
MATRIX HAS DIMENSIONS (R,C)=( 2 , 2 )

ENTER THE NAME FOR VARIABLE  3  -->  VAR. 3..
PLEASE ENTER: VALUE OF ROW  1  COLUMN 3 ? 50
PLEASE ENTER: VALUE OF ROW  2  COLUMN 3 ? 30_
```

Similarly, selections #11 and #12 add a column or row.

```
COL:     ONE         THREE        VAR. 3
-----------------------------------------
R 1:     13         -99.99          50
R 2:     1005        92             30
```

Result of column addition.

Selection #13 creates a new row or column with the same value.

```
ADD A (1) ROW OR (2) COLUMN WITH FIXED VALUE? 2
WHAT IS THE NEW VALUE? -1

ENTER THE NAME FOR VARIABLE  4  -->  FOUR....

COL:     ONE         THREE        VAR. 3         FOUR
-------------------------------------------------------
R 1:     13         -99.99          50            -1
R 2:     1005        92             30            -1
```

Result of selection #13.

```
CHANGE PORTIONS OF A (1) ROW OR (2) COLUMN? 1
WHAT IS THE NEW VALUE? 0
WHICH ROW/COLUMN IS TO BE CHANGED?  1
BEGIN CHANGE AT LOCATION? 1
END CHANGE AT LOCATION? 3
```

Selection #14 changes any sequence of row or column locations.

Save this matrix to disk by calling it "TEMP".

COL:	ONE	THREE	VAR. 3	FOUR
R 1:	0	0	0	-1
R 2:	1005	92	30	-1

```
THE CORE (A) MATRIX HAS DIMENSIONS -->  2 BY 4
THE SECOND (B) MATRIX HAS DIMENSIONS -->  2  BY 3

MERGE A WITH B BY (1) ROWS OR (2) COLUMNS --> ? 1
```

Selection #15 merges two matrices. To merge TEMP with TEST, enter selection 15 with TEMP in memory as the core matrix and then load TEST when prompted.

COL:	ONE	VAR. 02	THREE	FOUR
R 1:	0	0	0	-1
R 2:	1005	92	30	-1
R 3:	13	67	1.34	0
R 4:	1005	43	92	0

The maximum row and column dimensions are used for the merged matrix, and zeroes are used to fill in the empty places (as in locations 3,4 and 4,4).

```
THE MATRIX IN MEMORY HAS  4  ROWS AND  4  COLUMNS.

ENTER THE ROW, COLUMN COORDINATES OF THE UPPER LEFT
CORNER OF THE NEW MATRIX (R,C) --> ? 2,2

ENTER THE ROW, COLUMN COORDINATES OF THE LOWER RIGHT
CORNER OF THE NEW MATRIX (R,C) --> ? 4,3
```

Selection 16 allows a matrix to be split to create a new matrix. With the above merged matrix in memory, this procedure yields the results at bottom.

COL:	VAR. 02	THREE
R 1:	92	30
R 2:	67	1.34
R 3:	43	92

Select #1 to return to the main menu and then #11 for matrix multiplication.

NORTHERN HARDWOODS LABORATORY, U.S. FOREST SERVICE

MENU FOR: MATRIX MULTIPLICATION ROUTINE

1. RETURN TO MAIN MENU	2. GOTO BASIC
3. READ A MATRIX FROM DISK	4. WRITE A MATRIX TO DISK
5. ENTER A MATRIX MANUALLY	6. DISPLAY MATRIX ON VIDEO
7. PRINT MATRIX	8. MATRIX MULTIPLY

SELECT ONE OF THE ABOVE--> .

Enter data set TEST into memory and display it. Then select #8.

COL:	ONE	VAR. 02	THREE
R 1:	13	67	1.34
R 2:	1005	43	92

THE ORDER OF MULTIPLICATION IS B * A .

MATRIX B IS ALREADY IN MEMORY.

Remember: data set A must be conformable under the laws of matrix multiplication.

ENTER A FROM (1) DISK (2) KEYBOARD OR (3) IN MEMORY --> Then select #2 and enter the 3x1 matrix displayed below.

COL:	ONE
R 1:	1
R 2:	1
R 3:	1

After the multiplication routine, display the matrix in memory, the results of the matrix multiplication operation.

COL:	1
R 1:	81.34
R 2:	1140

Next, return to the main menu and select #13.

NORTHERN HARDWOODS LABORATORY, U.S. FOREST SERVICE

MENU FOR: MATRIX INVERSION ROUTINE

1. RETURN TO MAIN MENU
2. GOTO BASIC
3. READ A MATRIX FROM DISK
4. WRITE A MATRIX TO DISK
5. ENTER A MATRIX MANUALLY
6. DISPLAY MATRIX ON VIDEO
7. PRINT MATRIX
8. MATRIX INVERSION

SELECT ONE OF THE ABOVE--> .

Enter the following matrix into memory via #5 and select #8.

COL:	ONE	TWO	THREE
R 1:	2	1	1
R 2:	1	2	1
R 3:	1	1	2

If the matrix has a true inverse, the determinant will be displayed.

THE DETERMINANT = 4

PLEASE DEPRESS ANY KEY TO RETURN TO MENU.

The inverse of the matrix is in memory and may be displayed, printed, or saved to disk. Note that the column names are destroyed because they are no longer significant.

COL:	1	2	3
R 1:	.75	-.25	-.25
R 2:	-.25	.75	-.25
R 3:	-.25	-.25	.75

Next, return to the main menu and select #14.

```
                NORTHERN HARDWOODS LABORATORY, U.S. FOREST SERVICE

                MENU FOR:  ELEMENTARY ROW/COLUMN OPERATIONS ROUTINE

 1.  RETURN TO MAIN MENU                    2.  GOTO BASIC
 3.  READ A MATRIX FROM DISK                4.  WRITE A MATRIX TO DISK
 5.  ENTER A MATRIX MANUALLY                6.  DISPLAY MATRIX ON VIDEO
 7.  PRINT MATRIX                           8.  FLIP TWO ROWS
 9.  FLIP TWO COLUMNS                      10.  ROW * SCALAR
11.  COLUMN * SCALAR                       12.  ROW + SCALAR * ROW
13.  COL + SCALAR * COL                    14.  (ROW OR COL) / ELEMENT

 SELECT ONE OF THE ABOVE-->
```

Next, enter data set TEST into memory.

```
COL:       ONE          VAR. 02          THREE
-------------------------------------------------------------
R 1:       13             67              1.34
R 2:       1005           43              92
```

This routine defines the three allowable matrix elementary operations to be performed. In addition, you may exchange rows and columns. Select #8.

```
PLEASE ENTER: ROWS TO BE INTERCHANGED (X,Y)? 1,2_
```

Display the new data set with rows one and two interchanged.

```
COL:       ONE          VAR. 02          THREE
-------------------------------------------------------------
R 1:       1005           43              92
R 2:       13             67              1.34
```

Select #12 to add a constant times one row to another row. In the example, row #2 is multiplied by 2 and the result is added to row #1. The result is given below.

```
PLEASE ENTER:  R2 + K * R1    (R2,K,R1)? 1,2,2_
```

COL:	ONE	VAR. 02	THREE
R 1:	1031	177	94.68
R 2:	13	67	1.34

Select #14. This routine divides any row or column by any element identified by its location coordinates.

```
DO YOU WISH TO OPERATE ON A ROW (R) OR A COLUMN (C)? C

PLEASE ENTER:  WHICH ROW/COL DO YOU WISH TO OPERATE ON? 2

GIVE COORDINATES OF ELEMENT THAT WILL BE THE DIVISOR

(I,J)? 2,2_
```

COL:	ONE	VAR. 02	THREE
R 1:	1031	2.64179	94.68
R 2:	13	1	1.34

```
                NORTHERN HARDWOODS LABORATORY, U.S. FOREST SERVICE

                MENU FOR:  MATRIX DATA TRANSFORMATION ROUTINE

 1.  RETURN TO MAIN MENU                 2.  GOTO BASIC
 3.  READ A MATRIX FROM DISK             4.  WRITE A MATRIX TO DISK
 5.  ENTER A MATRIX MANUALLY             6.  DISPLAY MATRIX ON VIDEO
 7.  PRINT MATRIX                        8.  ABSOLUTE VALUE
 9.  INTEGER VALUE                      10.  EXPONENTIATION
11.  POWERS                             12.  SQUARE ROOT
13.  NATURAL LOGARITHM                  14.  BASE 10 LOGS
     SMOOTH VARIABLES

 SELECT ONE OF THE ABOVE--> ..
```

Enter the data set TEST into memory and display it. Then select #13.

```
COL:       ONE          VAR. 02          THREE
-------------------------------------------------------------
R 1:       13           67               1.34
R 2:       1005         43               92
```

Display the result. The natural logarithms of the original data set are now in memory. All other transformations work the same way.

Return to the main menu and select #16.

```
COL:       ONE          VAR. 02          THREE
-----------------------------------------------
R 1:       2.56495      4.20469          .29267
R 2:       6.91274      3.7612           4.52179
```

(END)

PROGRAMMING PROCEDURES

System Characteristics

NIMS is written in BASIC for the TRS-80 Model I computer. Two mini-disk drives and a lineprinter are recommended for best results. These programs should be easy to modify for any microcomputer system that uses the BASIC language.

Memory size determines the maximum dimensions of the data matrices with which the system can work. The current matrix dimensions for each major program of NIMS running on the fully expanded, 48,000 byte TRS-80 Model I system are:

a. MAIN/BAS	150 rows by 20 columns
b. ALTER/BAS	250 rows by 20 columns
c. MATMULT/BAS	45 rows by 45 columns
d. INVERSE/BAS	50 rows by 50 columns
e. TRNSFORM/BAS	150 rows by 20 columns
f. ELEMOPS/BAS	150 rows by 20 columns

The dimension statement, which defines size limit of the data set, is always located in line number 40000. Reducing the number of rows in a matrix increases the number of columns that can be accommodated and vice versa.

Data File Structure

The numerical data of interest are built into a matrix, Aij, where i = number of observations and j = number of variables. Each data set has three data files automatically defined for it. If "TEST" is the name of an example data set, then the data files "TESTM," "TESTN," and "TESTL" will be automatically created or updated every time "TEST" is saved to disk. File "TESTM" contains the matrix Aij, file "TESTN" contains the row and column dimensions, and file "TESTL" contains the names of the variables, i.e., columns, of Aij.

Program Code Documentation

Introduction

The programs are coded in the Microsoft (TM) BASIC language for direct use with the Radio Shack (TM) Level II BASIC interpreter. Unlike many other languages, BASIC allows concatention of several logical statements in one physical line. Microsoft BASIC uses the colon, ":," as the logical statement delimiter. Because each physical line costs 5 bytes of memory and each unnecessary space, 1 byte, small computer BASIC programs are written in compressed code. They look impossible to decipher, especially for those who normally program in FORTRAN. To make matters worse, internal program documentation is reduced to a minimum in BASIC because these statements use previous memory. Because of these memory constraints and the ability to refer unambiguously to any physical line in the program, the code is presented in the Appendix in the normal, compressed fashion. Each section of code is explained by reference to physical linenumbers.

Several conventions have been defined for the routines in this system. First, matrix A(R,C) always carries the data that is moved to and from disk, displayed on the video screen, or printed to the lineprinter. User-developed routines may use any variable names in computations as long as the results of data matrix operations end up in matrix A(R,C) before control is given to the input/output routines. Second, all missing values must be coded -99.99 to distinguish then from zeroes. The only program that now deals directly with missing values is program TRNSFORM. In general, operations that merely manipulate the structure of the data matrix will not be adversely affected by missing values. But you should be cautious: those routines that perform arithmetic operations on data elements are likely to produce errors if missing data are present.

NIMS is a group of 6 separate programs: MAIN, ALTER, MATMULT, INVERSE, TRNSFORM, and ELEMOPS. Limited memory necessitates balancing program length with data matrix size because both must reside in memory concurrently.

Program SEED (Appendix I)

The SEED program, lines 13-668, which is the core of all NIMS programs, is essentially the same code that is documented elsewhere by Rauscher and Buhyoff (1981). SEED is composed of 7 standard subroutines that control the input and output operations of other programs. These routines contain most of the TRS-80 specific statements. Users wishing to adapt NIMS to a different computer system must replace these 7 routines with similar ones defined by the new system requirements.

Lines 13-34 define a formatted input routine. The variable FL, set in the calling program, contains the formatting information. FL > 0 allows both numeric and alpha digits to be input from the keyboard. The number of digits allowed is defined by the absolute value of FL. When FL < 0, only numeric digits, the negative sign, and the period are acceptable inputs. When the absolute value of FL = 1, the input is accepted and processing continues without need for a carriage return. This is useful for rapidly accepting single digit choices such as "Y" or "2". At all other times, input will not be processed until a carriage return is entered from the keyboard. The string variable IN$ returns the input accepted to the calling program for further processing. You can substitute any routine that uses these variables to accomplish the same function.

The variable P, line 15, is set equal to the location of the screen cursor character. This address is stored in memory locations 16416 and 16417 (decimal) in the level II TRS-80 computer and must be changed to conform to the requirements of another computer system. Lines 16-23 pertain to single digit inputs and lines 24-34 pertain to multiple digit inputs.

Lines 36-47 implement a menu selection routine described in detail by Rowlett (1980). The address on the video screen where the blinking cursor is to appear must be passed to this subroutine from the main program in variable X. The number of items in the menu list must be passed in variable XO. On return to the calling program, X contains the item

number selected. Extensive use of the video screen standards specific to the TRS-80 makes this routine system dependent. It is used in NIMS only once—in program MAIN, line 10120—and could be replaced by a form of the input routine of lines 13-34.

The sequential data set disk-read routine is in lines 100-180 and the sequential data set write-to-disk routine is in lines 190-270. These two routines are system-specific. Before the write-to-disk routine is entered, the correct matrix of data, column names, and dimensions must be in the variables A(.,.), N$(.), R, and C, respectively. A substitute routine must be able to read and write from/to a disk the three files associated with each data set as discussed under "Data File Structure."

Routine input, lines 290-435, allows a user to enter a data set into the matrix structure from the keyboard. After specifying the data set dimensions, variables R and C, and the name of each column, variable N$(.), you enter the data elements one row at a time. This routine does not contain any particular system-dependent statements.

Routines video, lines 440-619, and lineprinter, lines 620-668, are essentially the same. The data matrix is displayed on the video screen or on a lineprinter. Before this routine is entered, the matrix must be in variable A(.,.) and the associated column names must be in N$(.) with the dimensions in R and C. Lines 510 and 623 act as an error trap to ensure that some matrix is in fact in memory. The video routine is designed for a screen of 16 rows by 64 columns. Such a screen allows 13 rows and 4 columns of the data set on display at any one time. Different row numbers and columns can be accommodated by changing the number 13 or 4 wherever it occurs between lines 440-619. These routines should be more or less universal.

Program MAIN (Appendix II)

The MAIN program gives you command control access to every other subprogram in NIMS. Line 10 reserves required string space, sets the dimensioning routine flag, TX, to ON = -1, and jumps to the menu routine, always located between lines 700-920. Subroutine 42000 accepts a string variable, T$, centers it, and prints it as the next line on the video screen. The routine in 40000 dimensions the data matrices required and initializes any variables that need it. Variable MV is used for missing values recognized as -99.99. The dimensioning routine may be accessed only once per execution because run-time redimensioning of data matrices is not permitted by most BASIC's. In line 750, TX is redefined as OFF = 1 and the menu is displayed. Keyboard input is accepted in line 900, and after an error trap, the appropriate subroutine is called in line 910. After return from each subroutine of MAIN, the central menu is once more displayed, line 920.

The addition and subtraction of matrices, scalar multiplication, matrix transposition, and creation of special matrices (menu options 9, 10, 12, & 15), respectively, are subroutines in program MAIN. Matrix alteration, multiplication, inversion, transformation, and elementary operations (menu options 8, 11, 13, 14, and 16), respectively, require the execution of other programs, lines 930-1010.

The matrix addition/subtraction subroutine, lines 6500-6700, stores the first matrix entered in variables B(.,.), RB, and CB, lines 6570-6602. The second matrix is retained in A(.,.), R, and C after it is entered into memory in line 6604. Conformation of matrices is tested in 6614, and the operation is performed, lines 6640-6670. If variable S = 1, subtraction is performed; in other cases, addition is done.

Scalar/matrix operations are performed in lines 6900-6950. Variable IT = 1 signifies multiplication and IT = 2 signifies addition (line 6932). The multiplier is stored in variable M and the scalar addition value is stored in variable AS (line 6935). Subtraction is done if AS is negative. In line 6940, you must respond to select the portion of the matrix targeted for the operation: row (lines 8000-8020), column (lines 8100-8120), or all elements (lines 8200-8210). The operation is performed in lines 8010, 8110, and 8200. An error check for missing values, MV, disallows operation on them.

Transposition is accomplished in lines 7500-7660. The only complicated part of this routine occurs in line 7645 which causes the now invalid column names to be changed into numeral names representing that column.

Finally, the routine to enter special matrices is defined in lines 10100-10190. This routine allows creation of diagonal matrices by entering only the diagonal values instead of every element (lines 10140-10190). Matrices in which all elements are the same number may also be created in this routine (lines 10130-10135).

Program ALTER (Appendix III)

Through ALTER, you can change an element of the data matrix (lines 4200-4260), delete a column

(lines 4500-4610), delete a row (lines 4800-4910), add a column (lines 5100-5170), add a row (lines 5400-5470), create a new row/column with a fixed value (lines 6000-6080), set parts of a row/column to a fixed value (lines 6090-6180), merge two matrices (lines 7000-7310), and extract any portion of a matrix to create a new one (lines 8000-8130). These subroutines are extremely simple. It should be equally simple to add operations if necessary. As in the MAIN program, one matrix is always stored in A(.,.) with column names in N$(.). A second matrix, if needed, is stored in B(.,.) and its column names are stored in NB$(.).

Program MATMULT (Appendix IV)

This program allows matrix multiplication (lines 7300-7480). It is a separate program in order to allow the largest possible matrices to be multiplied together. There is an error check (line 7345) for conformability of matrices under multiplication. The original column names no longer apply after multiplication (line 7470).

Program INVERSE (Appendix V)

This inversion routine (lines 8500-9526) was adapted from the program presented by Hornbeck (1975). The routine uses Gauss-Jordan elimination with column shifting to maximize pivot elements. The matrix to be inverted must be square, nonsingular, and of order 50 or less. The only error trap is in line 9312, which checks to see if the determinant of the matrix is zero to within six digits. This criterion could easily be changed. The Euclidean norm of the matrix is computed and stored in PD; DTNRM holds the value for the determinant, DETM, divided by PD. Extreme ill-conditioning or singularity will usually be accompanied by a small value of DTNRM (Hornbeck 1975).

Program TRNSFORM (Appendix VI)

Program TRNSFORM allows data elements to be transformed by various functions (lines 10000-10640). Many other transformation functions are possible and can easily be inserted. Transformations may now only be performed on the entire data matrix, not on selected rows or columns.

Program ELEMOPS (Appendix VII)

Program ELEMOPS allows elementary row/column operations plus the interchanging of any two rows or columns. The code to accomplish these tasks is clear and needs no additional documentation. The loop in line 10000 is used as a time delay loop.

Recovery From Errors

The most serious errors will terminate the program. The most effective recovery technique is simply to enter <GOTO 700>, re-activating the program by switching control to the current menu. The data containing the work done up to the error will still be active in memory as can be seen by entering option <6> to display it. You can save the current work, take stock of the situation, and try again. Although this procedure does not correct the error, it does save your time. Most errors will probably result from a conflict between the maximum dimensions allowed as defined in line 40000 and the current requirements. The solution is to adjust the dimensions of line 40000 if they differ from the current needs.

APPENDIX I. Source code listing for program SEED/BAS

```
13 '*************************************************
14 'ANSWER INPUT ROUTINE
15 P=(256*PEEK(16417)+PEEK(16416))-15359
     :IF ABS(FL)=1THEN16ELSE24
16 IN$=""
     :PRINT@P,CHR$(136);
     :LX=1
17 IN$=INKEY$
     :IFIN$=""THEN22
18 IFFL=-1 AND ASC(IN$)>47 AND ASC(IN$)<58 THEN 21
19 IF FL=1 AND ASC(IN$)>64 AND ASC(IN$)<91 THEN 21
20 GOTO 22
21 PRINT@P,IN$
     :RETURN
22 LX=LX+1
     :IFL<16THEN17
23 PRINT@P," ";
     :FORLX=1TO25
     :NEXT
     :GOTO16
24 PRINT@P, CHR$(136);
     :IN$=""
     :LX=ABS(FL)
     :GOSUB34
25 A$=INKEY$
     :IFA$=""THEN25
26 IFA$=CHR$(13)THENPRINTCHR$(15);
     :RETURN
27 IF A$=CHR$(8)THENIFIN$<>""THENPRINTA$;
     :PRINTCHR$(136);CHR$(24);
     :IN$=LEFT$(IN$,LEN(IN$)-1)
     :LX=LX+1
     :GOTO25
     :ELSEGOTO28
28 IFA$=CHR$(24)THENPRINTCHR$(29);
     :GOTO24
29 IF FL>0 AND ASC(A$)>31 AND ASC(A$)<91 THEN 32
30 IF FL<0 AND ASC(A$)>=45 AND ASC(A$)<58 THEN 32
31 GOTO 25
32 IF LX=0 THEN 25
33 PRINT A$;
     :IN$=IN$+A$
     :LX=LX-1
     :GOTO25
34 FOR IX=1TOLX-1
     :PRINT CHR$(136);
     :NEXT
     :FORJX=1TOLX
     :PRINT CHR$(24);
     :NEXT
     :RETURN
36 '*****************************************
37 'MENU SELECTOR ROUTINE BY FRANK B. ROWLETT, JR.
38 '80 MICROCOMPUTING 1(11)
     :195  NOV. 1980
```

```
39 'X=STARTING LOCATION OF CURSOR X0=# ITEMS IN MENU
40 'ON OUTPUT X=ITEM # SELECTED USES X,X0,X1,X2,X3 AND B$
41 '//////////////////////////
42 X1=X
     :PRINT@960,CHR$(30);"[ MOVES INDICATOR UP, ";CHR$(92);" MOVES INDI
      CATOR DOWN, <ENTER> TO SELECT";
43 PRINT@X1,CHR$(143);
     :FORX2=0TO2
     :B$=INKEY$
     :IFB$=""THENNEXTX2
     :PRINT@X1,CHR$(94);
     :FORX2=0TO9
     :B$=INKEY$
     :IFB$=""THENNEXTX2
     :GOTO43
44 PRINT@X1," ";
     :X2=ASC(B$)
     :IFX2=91THEN45ELSEIFX2=10THEN46ELSEIFX2=13THENPRINT@X1,CHR$(94);
     :X=(X1-X)/64+1
     :RETURNELSE43
45 X3=0
     :X1=X1-64
     :IFX1<XTHENX1=X+(X0-1)*64
     :GOTO47ELSE47
46 X3=0
     :X1=X1+64
     :IFX1=X+X0*64THENX1=X
47 PRINT@X1,CHR$(143);
     :FORX2=0TO5
     :NEXTX2
     :PRINT@X1,CHR$(94);
     :FORX2=0TO19
     :NEXTX2
     :IFX3<2THENX3=X3+1
     :GOTO47ELSEX3=PEEK(14656)
     :PRINT@X1," ";
     :IFX3=8THEN45ELSEIFX3=16THEN46ELSE43
50 '************************************
52 'CHANGE VARIABLE NAMES
54 '///////////////
56 CLS
     :PRINT@448,"WHICH COLUMN'S NAME WILL BE CHANGED? --> ";
     :FL=-2
     :GOSUB15
     :XJ=VAL(IN$)
     :PRINT
     :PRINT"PLEASE ENTER NEW NAME --> ";
     :FL=8
     :GOSUB15
     :N$(XJ)=IN$
     :RETURN
100 '*****************************************
110 'READ SEQUENTIAL DATA SET FROM DISK
120 '////////////////////
130 CLS
135 PRINT@448,"ENTER FILENAME OF DESIRED DATA SET --> ";
```

```
     :8,"ENTER FILENAME OF DESIRED DATA SET --> ";
     :INPUT F$
     :FF$=F$+"N"
     :L$=F$+"L"
     :F$=F$+"M"
140 OPEN "I",1,FF$
     :INPUT#1,R,C
     :CLOSE1
     :'GET R,C VALUES
145 OPEN "I",1,L$
     :FORK=1TOC
     :INPUT#1,N$(K)
     :NEXTK
146 CLOSE
150 OPEN "I",1,F$
     :
160 FORI=1TOR
     :FORJ=1TOC
     :INPUT#1,A(I,J)
     :NEXTJ,I
170 CLOSE 1
180 RETURN
190 '************************************************
200 ' WRITE A SEQUENTIAL MATRIX TO DISK
210 '////////////////////
220 CLS
     :PRINT@448,"ENTER FILENAME OF NEW DATA SET --> ";
     :INPUT F$
     :PRINT
     :PRINT"ENTER DISK DRIVE # WHERE DATA SHOULD BE SAVED --> ";
     :INPUT DD$
     :FF$=F$+"N:"+DD$
     :L$=F$+"L:"+DD$
     :F$=F$+"M:"+DD$
225 FOR JQ=1TOC
     :IF LEN(N$(JQ))=0 THEN N$(JQ)=STR$(JQ)
226 NEXT
230 OPEN "O",1,FF$
     :PRINT#1,R;C;
     :CLOSE1
240 OPEN "O",1,F$
     :
250 FORI=1TOR
     :FORJ=1TOC
     :PRINT#1,A(I,J);
     :NEXTJ,I
260 CLOSE 1
265 OPEN"O",1,L$
     :FORI=1TOC
     :PRINT#1,N$(I);",";
     :NEXT
     :CLOSE
270 RETURN
290 REM ************************************************
295 REM                SUBROUTINE INPUT
```

```
300 '1.  THIS SUB READS FROM THE KEYBOARD
305 '2.  MATRIX A IS LOADED AND CAN BE SAVED ON DISK
325 '6.  14-APR-79
335 REM ************************************************
340 CLS
345 INPUT "PLEASE ENTER: NO. ROWS AND COLS (R,C).--> ";R,C
350 FL=8
     :PRINT
     :FOR J=1TOC
     :PRINT"ENTER THE NAME FOR VARIABLE ";J;" --> ";
     :GOSUB15
     :N$(J)=IN$
     :PRINT
     :NEXT
355 CLS
365 FOR I=1 TO R
370 PRINT "ROW ";I
375 FOR J=1 TO C
380 FL=-12
     :PRINT "PLEASE ENTER:  VALUE OF COLUMN ";J" --> ";
     :GOSUB 15
385 PRINT
390 A(I,J)=VAL(IN$)
395 NEXT J
400 NEXT I
430 RETURN
435 END
440 '*****************************************************
445 '          SUBROUTINE VIDEO
450 '  DISPLAYS MATRIX A(R,C),13 ROWS BY 4 COLUMNS AT ONCE
455 ' USES ARROWS IN ALL DIRECTIONS TO SEE DIFFERENT PAGES
460 ' OF DATA.  "CLEAR" WILL TERMINATE DISPLAY
500 '*****************************************************
510 CLS
     :IFR=0 OR C=0 THEN RETURN ELSE JJ=INT(C/4.1+1)
515 IFINT(R/13)=R/13THENII=R/13ELSEII=INT(R/13+1)
517 I=1
     :J=1
525      IF R-13*I>=0.0 THEN N=13 ELSE N=13+(R-13*I)
535      IF C-4*J>=0.0 THEN L=4 ELSE L=4+(C-4*J)
540 CLS
     :PRINT"COL:";
     :FOR KK=0TO(L-1)
     :PRINT TAB(14*KK+8) N$((KK+1)+4*(J-1));
545 NEXT KK
     :PRINT
550 PRINT"-------------------------------------------------------
      -------"
555 FOR M=1 TO N
560 PRINT "R";M+13*(I-1);":";
565 FOR K=1 TO L
570 A=A(M+13*(I-1),K+4*(J-1))
575 PRINT TAB(14*(K-1)+8) A;
580 NEXT K
     :PRINT
585 NEXT M
```

```
586 A$=INKEY$
     :IFA$=""GOTO586
587 IF ASC(A$)=31THEN619ELSEIFASC(A$)=9THEN588ELSE590
588 IFJ+1>JJTHENJ=1ELSEJ=J+1
589 GOTO 525
590 IF ASC(A$)=8THEN591ELSE593
591 IF J-1<=0THENJ=JJ ELSE J=J-1
592 GOTO 525
593 IF ASC(A$)=10THEN594ELSE596
594 IF I+1>IITHENI=1ELSEI=I+1
595 GOTO 525
596 IF ASC(A$)=91THEN597ELSE601
597 IF I-1<=0THENI=IIELSEI=I-1
598 GOTO 525
601 IF ASC(A$)=46 THEN GOSUB 50 ELSE 618
602 GOTO 525
618 GOTO586
619 RETURN
     :END
620 '*************************************************
622 'LINEPRINTER OUTPUT ROUTINE
623 IF R=0 OR C=0 THEN RETURN
624 AN$="DATA SET NAME --> "
     :CLS
     :PRINT@448,"PRESS <ENTER> WHEN PRINTER IS READY.";
     :INPUT AN
626 PRINT
     :PRINT"PLEASE ENTER NAME OF DATA SET --> ";
     :FL=20
     :GOSUB15
     :T$=AN$+IN$
     :PRINT
     :PRINT
     :GOSUB42000
     :LPRINT TAB(ZZ);IN$
     :LPRINT" "
     :LPRINT" "
630 JJ=INT(C/4.1+1)
     :II=INT(R/200.1+1)
632 FORI=1TOII
634 IFR-200*I>=0.0THENN=200ELSEN=200+(R-200*I)
636 FORJ=1TOJJ
638 IFC-4*J>=0.0THENL=4ELSE L=4+(C-4*J)
640 LPRINT"COL:";
     :FORKK=0TO(L-1)
     :LPRINT TAB(14*KK+8) N$((KK+1)+4*(J-1));
642 NEXT KK
644 LPRINT" "
646 LPRINT"-------------------------------------------------------
     --------"
648 FOR M=1TON
650 LPRINT "R";M+200*(I-1);":";
652 FOR K=1TO L
654 A=A(M+200*(I-1),K+4*(J-1))
656 LPRINT TAB(14*(K-1)+8) A;
658 NEXT K
```

```
660 LPRINT
662 NEXT M
664 LPRINT " "
     :LPRINT " "
     :LPRINT " "
666 NEXT J,I
668 RETURN
670 CLS
     :PRINT@448,"THE CURRENT DATA MATRIX IN MEMORY WILL BE LOST."
     :PRINT
     :PRINT"DO YOU WISH TO (S)AVE IT OR (C)ONTINUE --> ";
     :FL=1
     :GOSUB15
     :IFIN$="S"THENGOSUB190ELSEIFIN$="C"THEN680ELSE670
680 RETURN
40000 DIM A(150,20),N$(20),B(150,20)
40015 MV=-99.99
40020 RETURN
42000 CC=LEN(T$)
     :ZZ=INT(30-CC/2)
     :PRINT TAB(ZZ);T$
     :RETURN
     :'CENTERING ROUTINE
43000 END
```

APPENDIX II. Source code listing for program MAIN/BAS

```
10 CLEAR 500
     :TX=-1
     :GOTO700
     :'NUMERICAL INFORMATION MANIPULATION SYSTEM  02/04/81
695 RETURN
700 '*****************************************************
710 'GENERALIZED I/O SUBROUTINES FOR PROGRAM STARTERS
720 '////////////////////
735 CLS
     :T$="NORTHERN HARDWOODS LABORATORY, US FOREST SERVICE"
     :GOSUB42000
736 T$="////////////////////////////////////////////////////"
     :GOSUB42000
737 PRINT
738 T$="MENU FOR:   NUMERICAL INFORMATION MANIPULATION SYSTEM"
739 GOSUB42000
     :PRINT
740 IF TX=-1 THEN GOSUB 40000 ELSE 750
750 TX=1
     :PRINT"  1.  RETURN TO MAIN MENU          2.  GOTO BASIC
760 PRINT"  3.  READ A MATRIX FROM DISK      4.  WRITE A MATRIX TO
      DISK
770 PRINT"  5.  ENTER A MATRIX MANUALLY      6.  DISPLAY MATRIX ON
      VIDEO
780 PRINT"  7.  PRINT MATRIX                 8.  ALTER MATRIX
790 PRINT TAB(2) "9.  ADD/SUB TWO MATRICES";TAB(33) "10.  SCALAR/
     MATRIX OPERATIONS"
800 PRINT TAB(1) "11.  MATRIX MULTIPLY";TAB(33) "12.  TRANSPOSE A
      MATRIX"
810 PRINT TAB(1) "13.  INVERSE OF A MATRIX";TAB(33) "14.  TRANSFO
     RM DATA"
820 PRINT TAB(1) "15.  CREATE SPECIAL MATRICES";TAB(33) "16.  ROW
     /COL OPERATIONS"
900 PRINT
     :PRINT"SELECT ONE OF THE ABOVE --> ";
     :FL=-2
     :GOSUB15
     :AN=VAL(IN$)
     :IFAN<=0ORAN>16THEN740
905 IF R=0 OR C=0 THEN 910 ELSE IF AN=1 OR AN=2 OR AN=3 OR AN=5 O
     R AN=8 OR AN=11 OR AN=13 OR AN=14 OR AN=16 THEN GOSUB 670
910 ON AN GOSUB 695,43000,130,220,340,510,620,930,6500,6900,960,7
     500,980,990,10100,1010
920 GOTO 735
930 RUN"ALTER/BAS"
     :END
960 RUN"MATMULT/BAS"
     :END
980 RUN"INVERSE/BAS"
     :END
990 RUN"TRNSFORM/BAS"
     :END
1010 RUN"ELEMOPS/BAS"
     :END
6500 '*****************************************************
```

```
6504 '           SUBROUTINE  ADDSUB
6508 '1.  ADDS B+C OR SUBTRACTS B-C
6512 '2.  OUTPUTS RESULTANT AS A
6513 '////////////////////
6532 CLS
     :PRINT"THIS ROUTINE ADDS OR SUBTRACTS MATRICES."
     :PRINT
     :PRINT"PLEASE ENTER:  (A)DD MATRIX B+C OR (S)UBTRACT  B-C --> ";
     :FL=1
     :GOSUB15
     :IFIN$="A"THENS=1ELSEIFIN$="S"THENS=2ELSE6532
6534 PRINT
6536 PRINT"ENTER MATRIX B FROM (1) DISK (2) KEYBOARD OR (3) ALREA
      DY IN MEMORY --> ";
     :FL=-1
     :GOSUB15
     :AN=VAL(IN$)
     :ON AN GOSUB 130,340,6690
6560 '    LOAD FIRST MATRIX INTO B
6570 FOR I=1 TO R
6580 FOR J=1 TO C
6590 B(I,J)=A(I,J)
6600 NEXT J,I
6602 RB=R
     :CB=C
6604 PRINT
     :INPUT"PLEASE ENTER: C FROM 1=DISK, 2=KEYBOARD, 3=ALREADY LOADED";
      NN
     :ON NN GOSUB 130,340,6690
6614 IF RB<>R OR CB<>C THEN 6616 ELSE 6640
6616 PRINT"MATRICES DO NOT CONFORM TO ADDITION/SUBTRACTION."
     :GOTO 6500
6630 '      ADDITION ROUTINE
6640 FOR I=1 TO R
6650 FOR J=1 TO C
6652 IF S=1 THEN 6660
6654 A(I,J)=B(I,J)-A(I,J)
6656 GOTO 6670
6660 A(I,J)=A(I,J)+B(I,J)
6670 NEXT J,I
6680 PRINT
6690 RETURN
6700 FOR I=1TOR
     :FORJ=1TOC
     :A(I,J)=B(I,J)
     :NEXT J,I
     :RETURN
6900 '************************************************
6902 '      SUBROUTINE SCAMULT
6904 '1.  MULTIPLIES A MATRIX BY A SCALAR
6906 '2.  OUTPUTS MATRIX A AS A RESULTANT
6908 '3.  H.M. RAUSCHER     15-APR-79
6910 '4.  15-APR-79
6930 IF R=0 OR C=0 THEN RETURN
     :'CHECK FOR NO MATRIX IN MEMORY
```

```
6932 CLS
     :PRINT@448,"ENTER (M)ULTIPLY OR (A)DD/SUBTRACT BY A SCALAR VALUE -
      -> ";
     :INPUT A$
     :IFA$="M"THENIT=1ELSEIFA$="A"THENIT=2ELSEGOTO6932
6935 CLS
     :IFIT=1THENPRINT@448,"ENTER VALUE OF SCALAR MULTIPLIER --> ";
     :INPUTM
     :ELSEIFIT=2THENPRINT@448,"ENTER VALUE (+/-) OF SCALAR CONSTANT TO
      ADD/SUBTRACT --> ";
     :INPUTAS
     :ELSE6935
6940 CLS
     :PRINT@448,"OPERATE ON A SINGLE (R)OW, (C)OLUMN OR (A)LL ELEMENTS
      --> ";
     :INPUTA$
     :IFA$="R"THENGOSUB8000ELSEIFA$="C"THENGOSUB8100ELSEIFA$="A"THENGOS
      UB8200ELSE6940
6950 RETURN
7500 '*****************************************************
7502 '          SUBROUTINE TRANSPO
7504 '1.  TRANSPOSES A MATRIX
7506 '2.  OUTPUTS MATRIX A
7508 '3.  CALLS DISKREAD,KEYBOARD
7510 '4.  H.M. RAUSCHER          15-APR-79
7512 '5.  15-APR-79
7520 IF R=0 OR C=0 THEN RETURN
     :'CHECK FOR NO MATRIX IN MEMORY
7560 FOR I=1 TO R
7570 FOR J=1 TO C
7580 B(J,I)=A(I,J)
7590 NEXT J,I
7595 TE=R
     :R=C
     :C=TE
7600 '    LOAD RESULT FROM B TO A
7610 FOR I= 1 TO R
7620 FOR J=1 TO C
7630 A(I,J)=B(I,J)
7640 NEXT J
7641 NEXT I
7645 GOSUB7660
     :'STANDARDIZE COLUMN NAMES
7650 RETURN
7660 FOR I=1TOC
     :N$(I)=STR$(I)
     :NEXTI
     :RETURN
8000 CLS
     :PRINT@448,"ENTER ROW NUMBER TO OPERATE ON --> ";
     :INPUT I
     :IFI>RORI<1THEN8000ELSE8010
8010 FORJ=1TOC
     :IFA(I,J)=MVTHEN8015ELSEIFIT=1THENA(I,J)=M*A(I,J)ELSEA(I,J)=AS+A(I
      ,J)
```

```
8015 NEXTJ
8020 RETURN
8100 CLS
     :PRINT@448,"ENTER COLUMN NUMBER TO OPERATE ON --> ";
     :INPUTJ
     :IFJ>CORJ<1THEN8100ELSE8110
8110 FORI=1TOR
     :IFA(I,J)=MVTHEN8115ELSEIFIT=1THENA(I,J)=M*A(I,J)ELSEA(I,J)=AS+A(I
      ,J)
8115 NEXT I
8120 RETURN
8200 FORI=1TOR
     :FORJ=1TOC
     :IFA(I,J)=MVTHEN8205ELSEIFIT=1THENA(I,J)=M*A(I,J)ELSEA(I,J)=AS+A(I
      ,J)
8205 NEXTJ,I
8210 RETURN
10100 '*****************************************************
10105 'SUBROUTINE TO ENTER SPECIAL MATRICES
10110 '////////////////////
10120 CLS
     :PRINT@448,"WHICH TYPE OF DATA MATRIX DO YOU WANT TO CREATE?"
     :PRINT@582,"DIAGONAL MATRIX"
     :PRINT@646,"MATRIX WITH ALL ELEMENTS THE SAME VALUE"
     :X=580
     :X0=2
     :GOSUB36
     :IFX=1THEN10140
10130 CLS
     :PRINT@448,"ENTER THE ROW DIMENSION --> ";
     :FL=-3
     :GOSUB15
     :R=VAL(IN$)
     :PRINT
     :PRINT"ENTER THE COLUMN DIMENSION --> ";
     :GOSUB15
     :C=VAL(IN$)
     :PRINT
     :PRINT"ENTER THE VALUE FOR ALL ELEMENTS --> ";
     :FL=-12
     :GOSUB15
     :AA=VAL(IN$)
10135 PRINT
     :PRINT"COMPUTATIONS IN PROGRESS."
     :FORJ=1TOC
     :N$(J)=STR$(J)
     :FORI=1TOR
     :A(I,J)=AA
     :NEXTI,J
     :RETURN
10140 CLS
10145 PRINT"LOAD INTO MEMORY A 1XC ROW MATRIX OF DIAGONALS."
10147 PRINT
10150 PRINT"ENTER MATRIX FROM (1) DISK (2) KEYBOARD OR (3) IN MEM
      ORY --> ";
```

```
     :ENTER MATRIX FROM (1) DISK (2) KEYBOARD OR (3) IN MEMORY --> ";:F
      L=-1:GOSUB15:AN=VAL(IN$):ON AN GOSUB 130,340,10190
10152 IF R<>1 PRINT
     :PRINT"MATRIX IS NOT 1XC."
     :PRINT
     :GOTO10145
10155 R=C
     :FOR I=1 TOR
     :FOR J=1 TO C
10160 B(I,J)=0
     :IF I=J THEN B(I,J)=A(1,J)
10165 NEXT J
10170 NEXT I
10175 '     LOAD MATRIX A
10180 FOR I=1 TO R
     :FOR J=1 TO C
     :A(I,J)=B(I,J)
     :NEXT J
     :NEXT I
10190 RETURN
40000 DIM A(150,20),N$(20),B(150,20)
40015 MV=-99.99
40020 RETURN
42000 CC=LEN(T$)
     :ZZ=INT(30-CC/2)
     :PRINT TAB(ZZ);T$
     :RETURN
     :'CENTERING ROUTINE
43000 END
```

APPENDIX III. Source code listing for program ALTER/BAS

```
10 CLEAR 500
     :TX=-1
     :GOTO700
     :'ALTER/BAS   VERSION 03/17/81
695 RUN"MAIN/BAS"
     :END
700 '*************************************************
710 'GENERALIZED I/O SUBROUTINES FOR PROGRAM STARTERS
720 '//////////////////
735 CLS
     :T$="NORTHERN HARDWOOD LABORATORY, US FOREST SERVICE"
     :GOSUB42000
736 T$="////////////////////////////////////////////////////////"
     :GOSUB42000
     :PRINT
739 T$="MENU FOR:      MATRIX ALTERATION ROUTINE"
     :GOSUB42000
     :PRINT
740 IF TX=-1 THEN GOSUB 40000 ELSE 750
750 TX=1
     :PRINT"  1.  RETURN TO MAIN MENU          2.  GOTO BASIC
760 PRINT"  3.  READ A MATRIX FROM DISK      4.  WRITE A MATRIX TO
      DISK
770 PRINT"  5.  ENTER A MATRIX MANUALLY      6.  DISPLAY MATRIX ON
      VIDEO
780 PRINT TAB(2) "7.  PRINT MATRIX";TAB(34) "8.  CHANGE AN ELEMEN
    T"
790 PRINT TAB(2)"9.  DELETE A COLUMN";TAB(33)"10.  DELETE A ROW"
800 PRINT TAB(1)"11.  ADD A COLUMN";TAB(33)"12.  ADD A ROW"
810 PRINT TAB(1)"13.  ADD FIXED VALUE ROW/COL";TAB(33)"14.  CHANG
    E R/C TO FIXED VAL."
820 PRINT TAB(1)"15.  MERGE TWO MATRICES";TAB(33)"16.  SPLIT A MA
    TRIX"
900 PRINT
     :PRINT"SELECT ONE OF THE ABOVE --> ";
     :FL=-2
     :GOSUB15
     :AN=VAL(IN$)
905 IF R=0 OR C=0 THEN 910 ELSE IF AN=1 OR AN=2 OR AN=3 OR AN=5 T
    HEN GOSUB 670
910 ON AN GOSUB 695,43000,130,220,340,510,620,4200,4500,4800,5100
    ,5400,6000,6090,7000,8000
920 GOTO 720
4200 '      ALTER ONE ELEMENT OF A MATRIX
4201 CLS
4210 INPUT "PLEASE ENTER: LOCATION OF ELEMENT TO BE CHANGED (ROW,
     COL).";M,N
4220 PRINT
4230 PRINT "MATRIX (R,C)= (";M;",";N;")=";A(M,N)
4240 PRINT
4250 INPUT "PLEASE ENTER:  NEW VALUE OF ELEMENT.";Z
     :A(M,N)=Z
4260 GOTO 5900
4500 '      DELETE ONE COLUMN
4501 CLS
```

```
4510 INPUT"PLEASE ENTER:  COLUMN TO BE DELETED.";M
4512 IF M>C PRINT"SORRY. COLUMN OUTSIDE OF RANGE." ELSE 4520
4514 GOTO 4510
4520 IF M < C GOTO 4540
4530 C=C-1
     :GOTO 4600
4540 FOR J1=M TO C
4550 FOR I1=1 TO R
4560 IF J1=C GOTO 4565 ELSE GOTO 4570
4565 C=C-1
     :GOTO 4600
4570 A(I1,J1)=A(I1,J1+1)
4580 NEXT I1
4590 NEXT J1
4600 FOR J=MTOC
     :N$(J)=N$(J+1)
     :NEXT
     :GOTO5900
4610 STOP
4800 '         DELETE ONE ROW
4801 CLS
4810 INPUT"PLEASE ENTER:  ROW TO BE DELETED.";M
4812 IF M>R PRINT"SORRY. OUTSIDE OF RANGE." ELSE 4820
4814 GOTO 4810
4820 IF M < R GOTO 4840
4830  R=R-1
     :GOTO 4900
4840 FOR I=M TO R
4850 FOR J=1 TO C
4860 IF I=R GOTO 4830
4870 A(I,J)=A(I+1,J)
4880 NEXT J
4890 NEXT I
4900 GOTO 5900
4910 STOP
5100 '        ADD A COLUMN
5101 CLS
5105 PRINT"MATRIX HAS DIMENSIONS (R,C)=(";R;",";C;")"
5107 FL=8
     :PRINT
     :PRINT"ENTER THE NAME FOR VARIABLE ";C+1;" --> ";
     :GOSUB15
     :N$(C+1)=IN$
     :PRINT
5110 FOR I=1 TO R
5120 PRINT"PLEASE ENTER: VALUE OF ROW ";I;" COLUMN ";C+1;
5130 INPUT A(I,C+1)
5140 NEXT I
5150 C=C+1
5160 GOTO 5900
5170 STOP
5400 '             ADD ONE ROW
5401 CLS
5402 PRINT"A ROW MAY NOT BE INSERTED BETWEEN OTHER ROWS."
5410 PRINT "MATRIX HAS DIMENSIONS (R,C)=(";R;",";C;")"
```

```
5420 FOR J=1 TO C
5430 PRINT"PLEASE ENTER: VALUE OF ROW";R+1;"COLUMN";J;
5440 INPUT A(R+1,J)
5450 NEXT J
5460 R=R+1
     :GOTO 5900
5470 STOP
5900 RETURN
     :'TO MAIN MENUE
6000 '  CREATE A NEW ROW OR COLUMN WITH FIXED VALUE
6010 CLS
     :INPUT"ADD A (1) ROW OR (2) COLUMN WITH FIXED VALUE";EX
6020 INPUT"WHAT IS THE NEW VALUE";NV
6025 IFEX=2THENPRINT
     :FL=8
     :PRINT"ENTER THE NAME FOR VARIABLE ";C+1;" --> ";
     :GOSUB15
     :N$(C+1)=IN$
     :PRINT
     :ELSE 6030
6030 IF EX=1 THEN EY=C
6032 IF EX=2 THEN EY=R
6040 FOR I=1TOEY
6050 IF EX=1 THEN A(R+1,I)=NV
6052 IF EX=2 THEN A(I,C+1)=NV
6060 NEXT I
6070 IF EX=1 THEN R=R+1
6072 IF EX=2 THEN C=C+1
6080 RETURN
6090 '  SET PARTS OF ROWS OR COLUMNS TO PARTICULAR VALUES
6100 CLS
     :INPUT"CHANGE PORTIONS OF A (1) ROW OR (2) COLUMN";EX
6110 INPUT"WHAT IS THE NEW VALUE";NV
6120 INPUT"WHICH ROW/COLUMN IS TO BE CHANGED";J
6130 INPUT"BEGIN CHANGE AT LOCATION";ES
6140 INPUT"END CHANGE AT LOCATION";EF
6150 FOR I=ES TO EF
6160 IF EX=1 THEN A(J,I)=NV
6162 IF EX=2 THEN A(I,J)=NV
6170 NEXT I
6180 RETURN
7000 '************************************************
7010 'ROUTINE TO MERGE TWO MATRICES
7020 'A IS STORED IN MEMORY ALREADY, B IS LOADED IN
7030 IF R=0 OR C=0 THEN RETURN
     :'CHECK FOR NO MATRIX IN MEMORY
7035 CLS
7040 'TRANSFER A-->B
7050 FORI=1TOR
     :FORJ=1TOC
     :B(I,J)=A(I,J)
     :NEXTJ,I
     :RA=R
     :CA=C
     :FORJ=1TOC
```

```
     :OR
     :FORJ=1TOC
     :B(I,J)=A(I,J)
     :NEXTJ,I
     :RA=R
     :CA=C
     :FORJ=1TOC
     :NB$(J)=N$(J)
     :NEXTJ
7060 PRINT@256,"THE CORE MATRIX IS IN MEMORY. READ THE SECOND MAT
      RIX FROM DISK."
7070 GOSUB135
     :RB=R
     :CB=C
7080 CLS
     :PRINT@448,"THE CORE (A) MATRIX HAS DIMENSIONS --> ";RA;" BY ";CA
7090 PRINT"THE SECOND (B) MATRIX HAS DIMENSIONS --> ";RB;" BY ";C
      B
7100 PRINT
     :INPUT"MERGE A WITH B BY (1) ROWS OR (2) COLUMNS --> ";IA
7110 IF IA=1THEN7120ELSEIFIA=2THEN7200ELSE7100
7120 C=FNMAX(CA,CB)
     :R=RA+RB
     :'ROW MERGE
7130 FORI=1TOR
     :FORJ=1TOC
7135 IF I<=RA AND J>CA THEN B(I,J)=0
7140 IF I>RA AND J<=CB THEN B(I,J)=A(I-RA,J)
7145 IF I>RA AND J>CB THEN B(I,J)=0
7150 NEXT J,I
7160 IF C=CA THEN 7300
7170 FOR J=CA+1 TO C
     :N$(J)=STR$(J)
     :NEXTJ
     :GOTO7300
7200 R=FNMAX(RA,RB)
     :C=CA+CB
     :'COLUMN MERGE
7210 FORJ=1TOC
     :FORI=1TOR
7215 IF J<=CA AND I>RA THEN B(I,J)=0
7220 IF J>CA AND I<=RB THEN B(I,J)=A(I,J-CA)
7225 IF J>CA AND I>RB THEN B(I,J)=0
7230 NEXTI,J
7240 FORJ=CA+1TOC+CB
     :NB$(J)=N$(J-CA)
     :NEXT
     :FORJ=1TOC
     :N$(J)=NB$(J)
     :NEXTJ
7300 FORI=1TOR
     :FORJ=1TOC
     :A(I,J)=B(I,J)
     :NEXTJ,I
7310 RETURN
```

```
8000 '*********************************************
8010 'ROUTINE TO EXTRACT ANY PORTION OF A MATRIX OUT
8020 'OF AN EXISTANT ONE AND DEFINE IT AS AN INDEPENDENT
8030 'MATRIX
8040 '////////////////////////////
8050 IF R=0 OR C=0 THEN RETURN
     :'CHECK FOR RESIDENCE OF A MATRIX
8060 CLS
     :PRINT@448,"THE MATRIX IN MEMORY HAS "R" ROWS AND "C" COLUMNS."
     :PRINT
     :PRINT"ENTER THE ROW,COLUMN COORDINATES OF THE UPPER LEFT"
     :PRINT"CORNER OF THE NEW MATRIX (R,C) --> ";
     :INPUT RU,CU
8070 IF RU<1 OR RU>R OR CU<1 OR CU>C THEN 8060
     :'ERROR CHECK
8080 PRINT
     :PRINT"ENTER THE ROW,COLUMN COORDINATES OF THE LOWER·RIGHT"
     :PRINT"CORNER OF THE NEW MATRIX (R,C) --> ";
     :INPUT RL,CL
     :IFRL<1 OR RL>R OR RL<RU OR CL<1 OR CL>C OR CL<CU THEN 8080
     :'ERROR CHECK
8090 FORJ=1TOC
     :NB$(J)=N$(J)
     :NEXT
     :'PRESERVE COLUMN NAMES
8100 FORI=RUTORL
     :FORJ=CUTOCL
     :B(I-RU+1,J-CU+1)=A(I,J)
     :NEXTJ,I
     :C=CL-CU+1
     :R=RL-RU+1
     :'EXTRACT NEW MATRIX
8110 FORJ=1TOC
     :N$(J)=NB$(CU+J-1)
     :NEXTJ
     :'RETAIN COLUMN NAMES
8120 FORI=1TOR
     :FORJ=1TOC
     :A(I,J)=B(I,J)
     :NEXTJ,I
     :'LOAD NEW MATRIX INTO A()
8130 RETURN
40000 DIM A(370,2),N$(20),B(370,2),NB$(20)
40010 MV=-99.99
40012 DEF FNMIN(A,B)=(A+B-ABS(A-B))/2
     :'MINIMUM OF A,B
40014 DEF FNMAX(A,B)=(A+B+ABS(A-B))/2
     :'MAXIMUM OF A,B
40020 RETURN
42000 CC=LEN(T$)
     :ZZ=INT(30-CC/2)
     :PRINT TAB(ZZ);T$
     :RETURN
     :'CENTERING ROUTINE
43000 END
```

APPENDIX IV. Source code listing for program MATMULT/BAS

```
10 CLEAR 500
     :TX=-1
     :GOTO700
     :'MATMULT/BAS  VERSION  02/04/81
695 RUN"MAIN/BAS"
     :END
700 '*************************************************
710 'GENERALIZED I/O SUBROUTINES FOR PROGRAM STARTERS
720 '////////////////////
735 CLS
     :T$="NORTHERN HARDWOODS LABORATORY, US FOREST SERVICE"
     :GOSUB42000
736 T$="////////////////////////////////////////////////////"
     :GOSUB42000
     :PRINT
739 T$="MENU FOR:     MATRIX MULTIPLICATION ROUTINE"
     :GOSUB42000
     :PRINT
740 IF TX=-1 THEN GOSUB 40000 ELSE 750
750 TX=1
     :PRINT"  1.  RETURN TO MAIN MENU          2.  GOTO BASIC
760 PRINT"  3.  READ A MATRIX FROM DISK      4.  WRITE A MATRIX TO
      DISK
770 PRINT"  5.  ENTER A MATRIX MANUALLY      6.  DISPLAY MATRIX ON
      VIDEO
780 PRINT TAB(2) "7.  PRINT MATRIX";TAB(34) "8.  MATRIX MULTIPLY"

900 PRINT
     :PRINT"SELECT ONE OF THE ABOVE --> ";
     :FL=-1
     :GOSUB15
     :AN=VAL(IN$)
905 IF R=0 OR C=0 THEN 910 ELSE IF AN=1 OR AN=2 OR AN=3 OR AN=5 T
     HEN GOSUB 670
910 ON AN GOSUB 695,43000,130,220,340,510,620,7300
920 GOTO 720
7300 '************************************************
7301 '        SUBROUTINE MATMULT
7302 '1.  MULTIPLIES MATRIX B*A
7303 '2.  CALLS SUB READISK,KEYBOARD
7304 '3.  OUTPUTS MATRIX A
7305 '4.  H.M. RAUSCHER        15-APR-79
7306 '5.  15-APR-79
7310 IF R=0 OR C=0 THEN RETURN
     :'CHECK OF MATRIX IN MEMORY
7320 CLS
     :PRINT@128,"THE ORDER OF MULTIPLICATION IS B * A."
     :PRINT
     :PRINT"MATRIX B IS ALREADY IN MEMORY."
7330 RB=R
     :CB=C
7332 FOR I=1TOR
     :FORJ=1TOC
     :B(I,J)=A(I,J)
     :NEXT J,I
```

```
7335 PRINT
     :PRINT
     :PRINT"ENTER A FROM (1) DISK (2) KEYBOARD OR (3) IN MEMORY --> ";
     :FL=-1
     :GOSUB15
     :AN=VAL(IN$)
     :ON AN GOSUB 130,340,7480
7340 '     TEST FOR CONFORMITY
7345 IF CB=R GOTO 7444 ELSE 7350
7350 PRINT"MATRICES NOT CONFORMABLE FOR MULTIPLICATION."
     :FOR I=1 TO 1000
     :NEXT I
     :RETURN
7442 '      MULTIPLICATION ROUTINE
7444 FOR I=1 TO RB
7446 FOR J=1 TO C
7448 S=0
7450 FOR K=1 TO CB
7452 S=S+B(I,K)*A(K,J)
7454 NEXT K
7456 C(I,J)=S
7458 NEXT J,I
7460 '       RELOAD C TO A
7462 FOR I=1 TO RB
7464 FOR J=1 TO C
7466 A(I,J)=C(I,J)
7468 NEXT J,I
7470 R=RB
     :FORI=1TOC
     :N$(I)=STR$(I)
     :NEXTI
7480 RETURN
40000 DM=45
     :DIM N$(DM),A(DM,DM),B(DM,DM),C(DM,DM)
40010 MV=-99.99
40020 RETURN
42000 CC=LEN(T$)
     :ZZ=INT(30-CC/2)
     :PRINT TAB(ZZ);T$
     :RETURN
     :'CENTERING ROUTINE
43000 END
```

APPENDIX V. Source code listing for program INVERSE/BAS

```
10 CLEAR 500
     :TX=-1
     :GOTO700
     :'INVERSE/BAS  VERSION 02/05/81
695 RUN"MAIN/BAS"
     :END
700 '*************************************************
710 'GENERALIZED I/O SUBROUTINES FOR PROGRAM STARTERS
720 '/////////////////
735 CLS
     :T$="NORTHERN HARDWOODS LABORATORY, US FOREST SERVICE"
     :GOSUB42000
736 T$="////////////////////////////////////////////////////"
     :GOSUB42000
     :PRINT
739 T$="MENU FOR:    MATRIX INVERSION ROUTINE"
     :GOSUB42000
     :PRINT
740 IF TX=-1 THEN GOSUB 40000 ELSE 750
750 TX=1
     :PRINT"  1.  RETURN TO MAIN MENU          2.  GOTO BASIC
760 PRINT"  3.  READ A MATRIX FROM DISK      4.  WRITE A MATRIX TO
      DISK
770 PRINT"  5.  ENTER A MATRIX MANUALLY      6.  DISPLAY MATRIX ON
      VIDEO
780 PRINT TAB(2) "7.  PRINT MATRIX";TAB(34) "8.  MATRIX INVERSION
     "
900 PRINT
     :PRINT"SELECT ONE OF THE ABOVE --> ";
     :FL=-1
     :GOSUB15
     :AN=VAL(IN$)
905 IF R=0 OR C=0 THEN 910 ELSE IF AN=1 OR AN=2 OR AN=3 OR AN=5 T
     HEN GOSUB 670
910 ON AN GOSUB 695,43000,130,220,340,510,620,8500
920 GOTO 720
8500 REM ***********************************************
8510 REM      SUBROUTINE FOR MATRIX INVERSION
8520 '  EMPLOYS GAUSS-JORDAN ELIMINATION WITH COLUMN
8530 '  SHIFTING TO MAXIMIZE PIVOT ELEMENTS.
8540 '
8550 '  INPUTS
     :
8560 '    A(I,I) MUST BE A SQUARE, NONSINGULAR MATRIX
8570 '    N = ORDER OF MATRIX
8580 '
8590 '  OUTPUTS
     :
8600 '    A(N,N) CONTAINS THE INVERSE
8610 '    DETM  CONTAINS THE DETERMINANT
8620 '    DTNRM  CONTAINS MAGNITUDE OF DET/EUCLIDIAN NORM
8630 '
8640 '  NOTE
     :  THIS ROUTINE TESTS FOR A SINGULAR MATRIX
8650 '         AND PRINTS AN ERROR OUT IF A(N,N) IS SINGULAR
```

```
8655 'SOURCE
     : HORNBECK, R.W. 1975. NUMERICAL METHODS. P.295
8660 REM ************************************************
8670 IF R=0 OR C=0 THEN RETURN
     :'CHECK FOR MATRIX IN MEMORY
8680 CLS
8700 IF R<>C GOTO 9500 ELSE N=R
8702 CLS
     :PRINT@ 520,"PLEASE BE PATIENT.  I AM COMPUTING."
8710 PD=1.0
8720 FOR L=1 TO N
8730 DD=0
8740 FOR K=1 TO N
8750 DD=DD+A(L,K)*A(L,K)
8760 NEXT K
8770 DD=SQR(DD)
8780 PD=PD*DD
8790 NEXT L
8800 DETM=1.0
8810 FOR L=1 TO N
8820 J(L+20)=L
8830 NEXT L
8840 FOR L=1 TO N
8850 CC=0.0
8860 M=L
8870 FOR K=L TO N
8880 IF((ABS(CC)-ABS(A(L,K))) > 0.0) GOTO 8910
8890 M=K
8900 CC=A(L,K)
8910 NEXT K
8920 IF L=M GOTO 9010
8930 K=J(M+20)
8940 J(M+20)=J(L+20)
8950 J(L+20)=K
8960 FOR K=1 TO N
8970 S=A(K,L)
8980 A(K,L)=A(K,M)
8990 A(K,M)=S
9000 NEXT K
9010 A(L,L)=1.0
9020 DETM=DETM*CC
9030 FOR M=1 TO N
9039 ON ERROR GOTO 9380
9040 A(L,M)=A(L,M)/CC
9050 NEXT M
9060 FOR M=1 TO N
9070 IF L=M GOTO 9140
9080 CC=A(M,L)
9090 IF CC=0.0 GOTO 9140
9100 A(M,L)=0.0
9110 FOR K=1 TO N
9120 A(M,K)=A(M,K)-CC*A(L,K)
9130 NEXT K
9140 NEXT M
9150 NEXT L
```

```
9160 FOR L=1 TO N
9170 IF J(L+20) = L GOTO  9290
9180 M=L
9190 M=M+1
9200 IF J(M+20) = L GOTO 9220
9210 IF N > M GOTO 9190
9220 J(M+20)=J(L+20)
9230 FOR K=1 TO N
9240 CC=A(L,K)
9250 A(L,K)=A(M,K)
9260 A(M,K)=CC
9270 NEXT K
9280 J(L+20)=L
9290 NEXT L
9300 DETM=ABS(DETM)
9310 DTNRM=DETM/PD
9311 CLS
9312 IF DETM=0.0 GOTO 9370
9320 PRINT"THE DETERMINANT = ";DETM
9321 PRINT
     :FORI=1TOC
     :N$(I)=STR$(I)
     :NEXTI
9322 GOTO 9435
9324 END
9370 '      ROUTINE TO HANDLE A SINGULAR MATRIX
9380 CLS
9390 PRINT"I AM SORRY.  MATRIX A  IS SINGULAR."
9400 PRINT"A SINGULAR MATRIX HAS NO UNIQUE INVERSE."
9410 PRINT"IF YOU MUST GET AN INVERSE,OBTAIN A PROGRAM"
9420 PRINT"WHICH CAN CALCULATE THE PSEUDO-INVERSE OF A"
9430 PRINT"SINGULAR MATRIX.  "
9435 PRINT
9436 PRINT"PLEASE DEPRESS ANY KEY TO RETURN TO MENUE."
9437 A$=INKEY$
     :IF A$<>"" RETURN ELSE 9437
9450 END
9500 '     ERROR DUE TO NON-SQUARE MATRIX
9510 PRINT"I AM SORRY.  MATRIX ";N4;" IS NOT SQUARE AND"
9520 PRINT"CONSEQUENTLY CANNOT BE INVERTED."
9522 PRINT"PLEASE DEPRESS ANY KEY TO CONTINUE."
9524 A$=INKEY$
     :IF A$="" THEN 9524
9526 RETURN
40000 DM=50
     :DIM A(DM,DM),B(DM,DM),N$(DM),J(120)
40010 MV=-99.99
40020 RETURN
42000 CC=LEN(T$)
     :ZZ=INT(30-CC/2)
     :PRINT TAB(ZZ);T$
     :RETURN
     :'CENTERING ROUTINE
43000 END
```

APPENDIX VI. Source code listing for program TRNSFORM/BAS

```
10 CLEAR 500
     :TX=-1
     :GOTO700
     :'TRNSFORM/BAS   VERSION 02/04/81
695 RUN"MAIN/BAS"
     :END
700 '*************************************************
710 'GENERALIZED I/O SUBROUTINES FOR PROGRAM STARTERS
720 '////////////////////
735 CLS
     :T$="NORTHERN HARDWOODS LABORATORY, US FOREST SERVICE"
     :GOSUB42000
736 T$="/////////////////////////////////////////////////"
     :GOSUB42000
     :PRINT
739 T$="MENU FOR:   MATRIX DATA TRANSFORMATION ROUTINE"
     :GOSUB42000
     :PRINT
740 IF TX=-1 THEN GOSUB 40000 ELSE 750
750 TX=1
     :PRINT"  1.  RETURN TO MAIN MENU          2.  GOTO BASIC
760 PRINT"  3.  READ A MATRIX FROM DISK      4.  WRITE A MATRIX TO
      DISK
770 PRINT"  5.  ENTER A MATRIX MANUALLY      6.  DISPLAY MATRIX ON
      VIDEO
780 PRINT TAB(2) "7.  PRINT MATRIX";TAB(34) "8.  ABSOLUTE VALUE"
790 PRINT TAB(2)"9.  INTEGER VALUE";TAB(33)"10.  EXPONENTIATION"
800 PRINT TAB(1)"11.  POWERS";TAB(33)"12.  SQUARE ROOT"
810 PRINT TAB(1)"13.  NATURAL LOGARITHM";TAB(33)"14.  BASE 10 LOG
      S"
900 PRINT
     :PRINT"SELECT ONE OF THE ABOVE --> ";
     :FL=-2
     :GOSUB15
     :AN=VAL(IN$)
903 IF AN>5 AND R=0 THEN 735
905 IF R=0 OR C=0 THEN 910 ELSE IF AN=1 OR AN=2 OR AN=3 OR AN=5 T
      HEN GOSUB 670
910 ON AN GOSUB 695,43000,130,220,340,510,620,10000,10040,10090,1
      0140,10200,10250,10300
920 GOTO 720
10000 FOR I=1 TO R
     :FOR J=1 TO J
10010 A(I,J)=ABS(A(I,J))
10020 NEXT J,I
10030 GOTO 10350
10040 ' TAKE INTEGER VALUES
10050 FOR I=1 TO R
     :FOR J=1 TOC
10060 A(I,J)=INT(A(I,J))
10070 NEXT J,I
10080 GOTO 10350
10090 '    EXPONENTIATE
10100 FOR I=1 TO R
     :FOR J=1 TO C
```

```
10110 A(I,J)=EXP(A(I,J))
10120 NEXT J,I
10130 GOTO 10350
10140 '     POWERS
10150 INPUT"PLEASE ENTER: POWER OF A[X";X
10160 FOR I=1 TO R
     :FOR J=1 TOC
10170 A(I,J)=A(I,J)[X
10180 NEXT J,I
10190 GOTO 10350
10200 '   SQUARE ROOT
10210 FOR I= 1TO R
     :FOR J=1 TO C
10220 A(I,J)=SQR(A(I,J))
10230 NEXT J,I
10240 GOTO 10350
10250 '   NATURAL LOGS
10260 FOR I=1 TO R
     :FOR J=1 TOC
10270 A(I,J)=LOG(A(I,J))
10280 NEXT J,I
10290 GOTO 10350
10300 '    BASE 10 LOGS
10310 FOR I=1 TO R
     :FOR J=1 TO C
10320 A(I,J)= LOG(A(I,J))/LOG(10)
10330 NEXT J,I
10350 RETURN
40000 DIM A(150,20),N$(20),B(150,20),SV(20)
40010 MV=-99.99
40020 RETURN
42000 CC=LEN(T$)
     :ZZ=INT(30-CC/2)
     :PRINT TAB(ZZ);T$
     :RETURN
     :'CENTERING ROUTINE
43000 END
```

APPENDIX VII. Source code listing for program ELEMOPS/BAS

```
10 CLEAR 500
     :TX=-1
     :GOTO700
     :'ELEMOPS/BAS    VERSION 02/04/81
695 RUN"MAIN/BAS"
     :END
700 '*************************************************
710 'GENERALIZED I/O SUBROUTINES FOR PROGRAM STARTERS
720 '////////////////////
735 CLS
     :T$="NORTHERN HARDWOODS LABORATORY, US FOREST SERVICE"
     :GOSUB42000
736 T$="////////////////////////////////////////////////////"
     :GOSUB42000
     :PRINT
739 T$="MENU FOR:   ELEMENTARY ROW/COLUMN OPERATIONS ROUTINE"
     :GOSUB42000
     :PRINT
740 IF TX=-1 THEN GOSUB 40000 ELSE 750
750 TX=1
     :PRINT"  1.  RETURN TO MAIN MENU          2.  GOTO BASIC
760 PRINT"  3.  READ A MATRIX FROM DISK      4.  WRITE A MATRIX TO
      DISK
770 PRINT"  5.  ENTER A MATRIX MANUALLY      6.  DISPLAY MATRIX ON
      VIDEO
780 PRINT TAB(2) "7.  PRINT MATRIX";TAB(34) "8.  FLIP TWO ROWS"
790 PRINT TAB(2)"9.  FLIP TWO COLUMNS";TAB(33)"10.  ROW * SCALAR"

800 PRINT TAB(1)"11.  COLUMN * SCALAR";TAB(33)"12.  ROW + SCALAR
      * ROW"
810 PRINT TAB(1)"13.  COL + SCALAR * COL";TAB(33)"14.  (ROW OR CO
      L) / ELEMENT"
900 PRINT
     :PRINT"SELECT ONE OF THE ABOVE --> ";
     :FL=-2
     :GOSUB15
     :AN=VAL(IN$)
     :CLS
903 IF AN>5 AND R=0 THEN 735
905 IF R=0 OR C=0 THEN 910 ELSE IF AN=1 OR AN=2 OR AN=3 OR AN=5 T
      HEN GOSUB 670
910 ON AN GOSUB 695,43000,130,220,340,510,620,1580,1660,1740,1800
      ,1860,1940,2100
920 GOTO 720
1580 '   INTERCHANGE TWO ROWS
1600 INPUT"PLEASE ENTER: ROWS TO BE INTERCHANGED (X,Y)";F,T
1610 FOR J=1 TO C
1620 TEMP=A(T,J)
     :A(T,J)=A(F,J)
     :A(F,J)=TEMP
1630 NEXT J
1640 GOTO 2000
1650 END
1660 '      INTERCHANGE ANY TWO COLUMNS
1680 INPUT"PLEASE ENTER: COLS TO BE INTERCHANGED (X,Y)";F,T
```

```
1690 FOR I=1 TO R
1700 TEMP=A(I,T)
     :A(I,T)=A(I,F)
     :A(I,F)=TEMP
1710 NEXT I
1715 TR$=N$(T)
     :N$(T)=N$(F)
     :N$(F)=TR$
1720 GOTO 2000
1730 END
1740 '       MULTIPLY A ROW TIMES A CONSTANT
1760 INPUT"PLEASE ENTER: ROW AND CONSTANT MULTIPLIER (R,K)";RR,K
1770 FOR J=1 TO C
     :A(RR,J)=K*A(RR,J)
     :NEXT J
1780 GOTO 2000
1790 END
1800 '     MULTIPLY A COLUMN BY A CONSTANT
1820 INPUT"PLEASE ENTER: COL AND CONSTANT MULTIPLIER (C,K)";CC,K
1830 FOR I=1 TO R
     :A(I,CC)=K*A(I,CC)
     :NEXT I
1840 GOTO 2000
1850 END
1860 '      ADDITION OF A ROW MULTIPLIED BY A K TO ANOTHER
1880 INPUT"PLEASE ENTER: R2 + K * R1   (R2,K,R1)";R2,K,R1
1890 FOR J=1 TO C
1900 A(R2,J)=A(R2,J)+K*A(R1,J)
1910 NEXT J
1920 GOTO 2000
1930 END
1940 '    ADDITION OF A COL MULTIPLIED BY A K TO ANOTHER
1960 INPUT"PLEASE ENTER: C2 + K * C1 (C2,K,C1)";C2,K,C1
1970 FOR I=1 TO R
1980 A(I,C2)=A(I,C2)+K*A(I,C1)
1990 NEXT I
2000 RETURN
2010 END
2100 REM*****************************************
2110 REM           ROUTINE TO DIVIDE A ROW/COL BY AN ELEMENT
2120 REM
2130 RR=1
     :INPUT"DO YOU WISH TO OPERATE ON A ROW (R) OR A COLUMN (C)";A$
2140 IF A$="C" THEN RR=2
2150 INPUT"PLEASE ENTER: WHICH ROW/COL DO YOU WISH TO OPERATE ON"
      ;K
2160 IF RR=2 GOTO  2190
2170 IF K>0 AND K<=R GOTO 2210
2180 PRINT"SELECTION ";K;"IS NOT A ROW. PLEASE TRY AGAIN."
     :PRINT
     :GOTO 2150
2190 IF K>0 AND K<=C GOTO  2210
2200 PRINT"SELECTION ";K;"IS NOT A COLUMN. PLEASE TRY AGAIN."
     :PRINT
     :GOTO 2150
```

```
2210 INPUT"GIVE COORDINATES OF ELEMENT THAT WILL BE THE DIVISOR (
     I,J)";A,B
2220 IF A>0 AND B>0 AND A<=R AND B<=C GOTO 2250
2230 PRINT"THE MATRIX HAS ";R;" ROWS AND ";C;" COLUMNS."
2240 PRINT"PLEASE SELECT I AND J WITHIN THESE BOUNDS."
     :PRINT
     :GOTO 2210
2250 TE=A(A,B)
     :IF RR=2 GOTO 2280
2260 FOR I=1 TO C
     :A(K,I)=A(K,I)/TE
     :NEXT I
2270 RETURN
2280 FOR I=1 TO R
     :A(I,K)=A(I,K)/TE
     :NEXT I
2290 RETURN
2300 END
10000 FOR II=1 TO 1000
     :NEXT II
     :RETURN
40000 DIM A(150,20),N$(20),B(150,20),NB$(20)
40015 MV=-99.99
40020 RETURN
42000 CC=LEN(T$)
     :ZZ=INT(30-CC/2)
     :PRINT TAB(ZZ);T$
     :RETURN
     :'CENTERING ROUTINE
43000 END
```


U.S. GOVERNMENT PRINTING OFFICE:1983-666-754/5043

Rauscher, Harald M.
The microcomputer scientific software series 1: the numerical information manipulation system. Gen. Tech. Rep. NC-84. St. Paul, MN: U.S. Department of Agriculture, Forest Service, North Central Forest Experiment Station; 1983. 44p.

The Numerical Information Manipulation System extends the versatility provided by word processing systems for textural data manipulation to mathematical or statistical data in numeric matrix form. Numeric data, stored and processed in the matrix form, may be manipulated in a wide variety of ways. The system allows operations on single elements, entire rows, or columns, or on entire data matrices. Data elements may be transformed by logarithmic, power, exponential, and other operations. When data sets are treated as a single matrix unit, they may be added, subtracted, multiplied, inverted, and separated or merged in any desired configuration.

KEY WORDS: Data analysis, TRS-80, data processing, matrix operations.